Login Check

Written by:
David L. Stevens
Mark Morrow

Written at: *Stevens Institute of Technology*

September 18, 1989

Contents

1 Functional Overview **2**

1.1 Abstract 2
1.2 Check Disk 2
1.3 Check Expiration 2
1.4 Check User 3
1.5 Check Node Access 3
1.6 Check Schedule 3
1.7 Set Process Name 4

2 Technical Overview **5**

2.1 Required Logical Names 5
2.2 Usage 6
2.3 Software Modules 8
2.4 Hardware and Software Requirements 10

3 Software Installation **11**

Chapter 1

Functional Overview

1.1 Abstract

LOGIN-CHECK is a utility program that was designed to verify certain important pieces of information before a user is allowed to log onto the system. These are checks that the VMS Operating System has left out. Due to the lack of hooks into the VMS Operating System, the program must be run in the system login procedure, SYLOGIN. To insure that no one can avoid the checking, it is recommended that all accounts be set with *NOCONTROL_Y*, and that you enable *Control Y* before exiting SYLOGIN. The LOGIN-CHECK program should be run at the beginning of the SYLOGIN procedure, so as to not waste resources if the job is going to be flushed.

1.2 Check Disk

This is a way of verifying that the login device that the user uses is actually MOUNTED before the user logs in. If the volume is not mounted, then the user receives a message and is logged out. Any account with either SYSPRV or OPER privileges will be ignored by this checking function.

This check has been provided to make sure that a user does not get onto the system when their login device is OFFLINE. This now allows the system staff to take a single user disk offline, for maintenance or backups, and not effect other users of the system.

Note: Users with both SYSPRV and OPER privileges will not be prevented from logging onto the system even if their login device is OFFLINE.

1.3 Check Expiration

This option will issue a warning message to a user if the program determines that their VAX account is about to expire. This gives the user time to take action to either have the account extended, or to clean up their account before the expiration date.

The VMS operating system provides the user with a one week warning if their password is about to expire, but it gives no notification that their account is about to expire. The Check Expiration option supplies that missing functionality, but unlike the Password expiration warning of one week,

this option allows the system manager to set the warning interval. The system manager can set the warning interval to any amount of time he wishes. The interval is determined by the value of the logical name CHECK$SIT_EXPIRE_WARNING, this logical name must be set up as a VMS delta time value.

1.4 Check User

The Check User option, is a way to limit users from logging on multiple times to a VAX system. If this option is enabled, it will only limit the number of INTERACTIVE logins, it will not effect the number of Batch, Network, or Detached jobs you can have on the system.

NOTE that to enable this function, the Login-Check program must be installed with WORLD privileges. This is required for you to check the other processes on the system.

This option has been set up to allow only One Interactive Job per Username per Node. The checking will also ignore any interactive job in a Resource Wait State, this is done incase a user has a job hung up, they can still login to the system.

If a user has the SECURITY privilege enabled by default, then they can login multiple times on the same node. This back door has been provided to allow the system staff access to the system.

1.5 Check Node Access

This is a check that could be very useful in either a heterogeneous cluster or a large Local Area VAXCluster (LAVC). In either case, it is possible that you have a centralized SYSUAF, but don't want all accounts to be accessible from all nodes in the cluster. This checking will verify that the account has been granted access to the node they are logging into. If you have not been granted access to the node you are logging into, then you will be logged out.

If this option is being used, then a Node Access database must be set up, and into this you must add an entry for each node a Username has access to. The Node Access database is pointed to by the CHECK$SIT_ACCESS_DATABASE logical name.

Granting, revoking, and viewing access by a user is accomplished via the NODE_ACCESS.COM procedure. This is a simple menu based command procedure that will allow you to check on a users node access rights.

1.6 Check Schedule

The SCHEDULE option is a way of letting users know what is happening on the system, as well as a way of preventing access to the system until the system is ready for users.

The first option, called SCHED_NO_OPERATOR, will at login time inform users that the system currently has no OPERATOR coverage. This will at least warn users that requests for Disk and Tape mounts can not be handled. The second part, called SCHED_STAND_ALONE, is a way to prevent users from logging in until you are ready for them. This allows the systems staff to boot up

an entire system fully loaded, but not allow users to log in. This option will cause an error message to be displayed, and then the user will be logged off the system.

Note: Any user with SECURITY privilege will be able to ignore the Stand Alone schedule. This will allow the Systems staff access to the system without letting normal users on.

1.7 Set Process Name

This is not actually a checking function, but is a convenience if you allow multiple logins on your system. The routine will set your process name to your *USERNAME* if this is the first login or *USERNAME;x*, where *x* is sequence number from 1 thru 99, for all subsequent logins. By default VMS will set your first process name to be your *USERNAME*, and all subsequent jobs to be your login terminal name.

Chapter 2

Technical Overview

This chapter will give a more detailed description of each section of the Login-Check program. It will detail how to set the software up, and how to enable the various options available.

2.1 Required Logical Names

The Login-Check program requires that several logical names be set up. The program requires that all logical names be defined as System logical names in the LNM$SYSTEM_TABLE. Below is a detailed description of each logical name required by the program.

- **CHECK$SIT_ENABLED**

  Determines what checking you are performing. By adding the Check values, you set up a bit mask that will be checked internally to the program. All values indicated are in decimal.

  1. Perform Check-Disk validation (value = 1)
  2. Perform Check-Expiration warning (value = 2)
  3. Perform Check-User validation (value = 4)
  4. Perform Check-Node validation (value = 8)
  5. Perform Change-Process-Name function (value = 16)
  6. Perform Schedule validation (value = 32)

- **CHECK$SIT_EXPIRE_WARNING**

  Determines the warning interval for announcing that an account is about to expire. The time must be in Delta-Time format, ie **xxx yy:yy:yy** where **xxx** is the number of Days, and **yy:yy:yy** is the number of hours.

- **CHECK$SIT_ACCESS_DATABASE**

  Pointer to the database that tells which accounts have been granted access to which nodes in a cluster. This logical name need only be defined if the Check Node Access option has been enabled.

- **CHECK$SIT_SCHEDULE**

  Determines the availability of the system. Currently the only valid settings are:

  - Sched_Normal (value = 1)
  - Sched_No_Operator (value = 10)
  - Sched_Stand_Alone (value = 400)

  These settings should only be modified by the program SET_SCHEDULE.EXE, which will be described later.

2.2 Usage

The Login-Check utility does not require any inputs on the command line. If you wish to enable the Check-User option, then you will have to INSTALL the software with *WORLD* privileges. This is required inorder to obtain information on processes other than your own. The program should be installed in the SYS$SYSTEM area, and is invoked with the following command:

$ RUN SYS$SYSTEM:LOGIN_CHECK.EXE

The utility NODE_ACCESS.COM is used to Grant, Deny, and View node access given to a user. This utility is only required when the CHECK_NODE_ACCESS option has been enabled. To run the command procedure, issue the following command at the command line:

$ @SYS$SYSTEM:NODE_ACCESS.COM

The menu that will be displayed will look like the following:

```
                Node Access Menu

          1)  Grant a User access to a Node
          2)  Deny a User access to a Node
          3)  View Access granted to a User
          4)  Exit

     * Please specify your option:
```

To choose an option, you must enter the corresponding number at the prompt.

The program SET_SCHEDULE.EXE is the program used to manipulate the Schedule setting. The program will take its input off the command line. To run the program first define a symbol for it as shown below:

$ SETSCHED :== SYSSYSTEM:SET_SCHEDULE.EXE

Then invoke the program like so:

$ SETSCHED xxx

Where **xxx** is one of the following:

- *nothing*

  If nothing is specified, the program will show the current setting of the system Schedule.

- **SHOW**

  Will also cause the current setting of the system Schedule to be displayed.

- **HELP**

  Will supply help text on how to use the program.

- **1**

  Set the system Schedule to normal access.

- **10**

  Set the system Schedule to No-Operator Coverage.

- **400**

  Set the system Schedule to Stand-Alone, do not let users log in.

2.3 Software Modules

This program is written entirely in VAXC. The code takes advantage of several VMS system service routines in order to obtain the information it needs. Below is a list of all required software modules as well as how to build the program:

- **CHECK_AT_LOGIN.C**

  Driver module for the Login-Check utility. This module will dispatch to the routines which perform the actual checking. The program will only call those routines that have been enabled by the CHECK$SIT_ENABLED logical name.

- **CHECK_DISK.C**

  This is the module that implements the Disk Checking procedure. The routines will verify that the users disk is actually ONLINE and MOUNTED before it will let the user log in.

- **CHECK_USER.C**

  This is the module that implements the User Checking procedure. The routines will not allow a second interactive job to be logged in on the same node by any user.

- **CHECK_SCHEDULE.C**

  Module that handles the checking of the Schedule setting. The routines will not allow interactive users to log on if the Schedule is set to Stand-Alone.

- **CHECK_EXPIRATION.C**

  This is the module that implements the Account Expiration Checking procedure. The routine will check to see if the account being logged into is about to expire. The warning interval is determined by the CHECK$SIT_EXPIRE_WARNING logical name.

- **CHECK_NODE_ACCESS.C**

  This is the module that implements the Node verification procedure. The routine will check to see if the user logging in has been granted access to the node that they are logging into. This checking should only be performed if you use a common SYSUAF, and don't want all accounts to work on all nodes in the cluster.

- **CHECK_PROCESS_NAME.C**

  This is the module that takes care of setting your process name. This function need not be enabled if you also enable the CHECK_USER option. The routine will set your process name to your *USERNAME* for the first login, and to *USERNAME;x*, where *x* is a sequence number, for all subsequent logins.

- **CHECK_FILE.C**

  This module contains the basic File I/O routines required by the various checking modules. This module gives an easy callable interface to the Record Management Services.

- **CHECK.H, UAIDEF.H, STATEDEF.H**

  These are the INCLUDE files required by the various modules.

- **DESCRIP.MMS**

  This is an MMS descriptor file, used for building the software.

- **DEBUG-DESCRIP.MMS**

  This is an MMS descriptor file, used for building a debug version of the software.

- **LOGIN_CHECK_STARTUP.COM**

  This is the Startup Command Procedure that should be called as part of your System Startup Procedures. This is where the Logical names are defined, and where the Program is installed into memory (if the CHECK_USER option is being used).

- **NODE_ACCESS.COM**

  This is a command procedure that is used to grant or deny access for a user on a particular node. The command procedure is menu driven, and allows a privileged person to Grant, Remove, or Show node access for a given Username.

- **CHECK_NODE_ACCESS.FDL**

  This File Descriptor file defines the database used by the CHECK_NODE_ACCESS option. This database is an indexed sequential file that contains USERNAMEs, and the NODEs on which they have been granted access.

- **SET_SCHEDULE.C**

  This is the program that is used to maniplate the Schedule setting. The program allows you to *Set, Modify,* or *Show* the Schedule value. The program will also supply help on how to use it if you enter *HELP* as the command arguement

To build the software, in the case that you do not have MMS on your system, use the following commands:

```
$
$ CC/OPTIMIZE/NODEBUG CHECK_AT_LOGIN.C
$ CC/OPTIMIZE/NODEBUG CHECK_DISK.C
$ CC/OPTIMIZE/NODEBUG CHECK_USER.C
$ CC/OPTIMIZE/NODEBUG CHECK_EXPIRATION.C
$ CC/OPTIMIZE/NODEBUG CHECK_NODE_ACCESS.C
$ CC/OPTIMIZE/NODEBUG CHECK_PROCESS_NAME.C
$ CC/OPTIMIZE/NODEBUG CHECK_FILE.C
$ CC/OPTIMIZE/NODEBUG CHECK_SCHEDULE.C
$ LINK/NODEBUG/NOTRACE/EXECUTABLE=LOGIN_CHECK.EXE -
    CHECK_AT_LOGIN, CHECK_DISK, CHECK_USER, CHECK_EXPIRATION -
    CHECK_NODE_ACCESS, CHECK_PROCESS_NAME, CHECK_FILE
$
$ CC/OPTIMIZE/NODEBUG  SET_SCHEDULE.C
$ LINK/NOTRACE/NODEBUG SET_SCHEDULE.C
$
```

To install the LOGIN-CHECK program into memory, the following command must be used:

$ INSTALL ADD SYS$SYSTEM:LOGIN_CHECK.EXE/PRIVILEGE=WORLD

2.4 Hardware and Software Requirements

This software has been tested on single and multiprocessor VAX/VMS computers. It requires version 5.1 or later of the VMS Operating System.

Chapter 3

Software Installation

The Login Check utility is not a VMSINSTALable product, you must move the needed utilities to the required directories by hand. To install the Login Check software you must follow the steps listed below:

1. **Restore the files from the Save Set**

   The software must be extracted from the Save Set and moved into a disk directory on your local system.

2. **Move LOGIN_CHECK.EXE to the System directory**

   The software should be moved to the SYS$SYSTEM directory and run from there. If you wish to install the software in another directory, then you will have to modify the LOGIN_CHECK_STARTUP.COM procedure to reference the correct device and directory.

3. **Move SET_SCHEDULE.EXE to the System directory**

   The software should reside in the SYS$SYSTEM directory and be run from there. If you install the software in another directory, you will have to modify the start up procedure to reference the correct directory.

4. **Move LOGIN_CHECK_STARTUP.COM to the Manager directory**

   The startup file must be executed at System Startup time in order to define the CHECK logical names when the system reboots. We suggest that you move the Startup file to the SYS$MANAGER: directory, and add the following line to your System Startup Procedure.

   $ @SYS$MANAGER:LOGIN_CHECK_STARTUP.COM

   The Login Check Startup procedure should be modified to turn on the checking options that you want enabled. By default the procedure already enables the Disk checking, and Account Expiration checking options.

5. **Create the Node Access Database**

   If you are not enabling the Check Node option, then you may skip this item. Use the CHECK_NODE_ACCESS.FDL file to create a Node Access database. The database should be located in the SYS$MANAGER: directory, and can be created with the following command:

```
$ CREATE/FDL=CHECK_NODE_ACCESS.FDL -
_$ SYS$MANAGER:CHECK_NODE_ACCESS.DAT
```

   If you wish to call the database anything else, or have it located in another directory, they you must modify the logical name definition in the LOGIN_CHECK_STARTUP.COM procedure.

6. **Update the SYLOGIN.COM**

   The Login Check utility should be the first line executed in the SYLOGIN.COM procedure. This will insure that checking is done before you define your user environment. It is also recommended that all accounts have the *NOCONTROL_Y* flag set, and that you enable *Control Y* before you exit the SYLOGIN procedure. This will insure that noone will be able to abort out of the checking program.

   All support staff members who will be manipulating the Schedule setting, must define the following symbol for their processes.

   SETSCHED :== SYSSYSTEM:SET_SCHEDULE.EXE

   This is required since the Set-Schedule program must read its arguement off the command line.