FastLogin Facility for OpenVMS

Thierry Lelégard

Digital Equipment France

Release 3
17 April 1996

Abstract

When a user logs into an OpenVMS system using DECnet or LAT (`SET HOST` and `SET HOST/LAT` commands), the normal login process applies. A user-name and password are required. In environments where remote connections are frequently used, the users may be rapidly fed up with providing a user-name and a password every time.

FastLogin allows registered *trusted remote users* to log into the system without having to supply a user-name or a password. Inside a VMScluster, all users may be implicitely authorized to log into any other node of the same VMScluster using the same user-name without having to supply their password

Contents

1. FASTLOGIN FEATURES 2

1.1 USING A DEFAULT LOGIN ACCOUNT 2
1.2 USING A LOGIN KEY 2

2. FASTLOGIN MANAGEMENT 3

2.1 DECLARING TRUSTED REMOTE DECNET USERS 3
2.2 ENABLING AUTOMATIC REMOTE DECNET LOGIN 4
2.3 DECLARING TRUSTED REMOTE LAT USERS 4
2.4 ENABLING AUTOMATIC REMOTE LAT LOGIN 4
2.5 FASTLOGIN INSIDE A VMSCLUSTER 5
2.6 ENABLING AUTOMATIC LOGIN INSIDE A VMSCLUSTER 5
2.7 MANAGING A CLUSTER ALIAS 6
2.8 VERSION RESTRICTIONS 6
2.8.1 FastLogin restriction on OpenVMS versions before 6.1 6
2.8.2 Restriction on OpenVMS Alpha version 6.1 6

3. INSTALLATION 7

3.1 REBUILDING FASTLOGIN 7
3.2 INSTALLING FASTLOGIN 7

4. FASTLOGIN TROUBLESHOOTING 8

4.1 TRACING LOGIN PROBLEM 8
4.2 SENDING FASTLOGIN TRACE TO A FILE OR OPCOM 9
4.3 DISABLING FASTLOGIN WITHOUT REBOOTING 10
4.4 DISABLING FASTLOGIN WHEN REBOOTING 10
4.5 LAST RESOURCE 10

1. FastLogin Features

1.1 Using a default login account

Using FastLogin, the system manager of NODE_A declares that NODE_B::USER_B is a trusted user and is authorized to log into system NODE_A using account USER_A without supplying a password.

Then, from node NODE_B, USER_B tries to log into NODE_A:

```
$ SET HOST NODE_A

Username [USER_A]:

   Welcome to OpenVMS Alpha Operating System, Version ...
    Last interactive login on ...
    Last non-interactive login on ...

$
```

When the connection request comes from USER_B on NODE_B, the default local user-name is displayed between square brackets. If the user simply hits the return key, the user is logged as USER_A without having to supply a password.

However, if the user wants to use another account, the normal login processing applies:

```
$ SET HOST NODE_A

Username [USER_A]: OTHER_USER
Password: password

   Welcome to OpenVMS Alpha Operating System, Version ...
    Last interactive login on ...
    Last non-interactive login on ...

$
```

1.2 Using a login key

When the default login is accepted, the user may specify a ***key***. This key is any character string. The key must be entered at the "Username" prompt. Instead of giving a user-name, type a sharp sign ("#") followed by the key string. Because of the sharp sign, the input will not be interpreted as a user-name and the default account will be used.

After the login, the key string is placed into the logical name FASTLOGIN_KEY and may be used by the process.

Example:

```
$ SET HOST NODE_A

Username [USER_A]: #My Key
```

```
    Welcome to OpenVMS Alpha Operating System, Version ...
     Last interactive login on ...
     Last non-interactive login on ...

$ SHOW LOGICAL FASTLOGIN_KEY
   "FASTLOGIN_KEY" = "My Key" (LNM$PROCESS_TABLE)
$
```

Typical usage:

This feature can be used to implement automatic login from command procedures with automatic selection of an application.

Example: Suppose that USER_B on NODE_B executes a DCL command procedure which proposes a menu. Some of the proposed actions must be executed on NODE_A with account USER_A.

The DCL command on NODE_B looks like:

```
$ ....
$ DO_OPTION_1:
$ SET HOST NODE_A
#OPTION_1
$ GOTO MENU
$!
$ DO_OPTION_2:
$ SET HOST NODE_A
#OPTION_2
$ GOTO MENU
$ ....
```

On NODE_A, the LOGIN.COM procedure of USER_A contains something like:

```
$ KEY = F$TRNLNM ("FASTLOGIN_KEY")
$ IF KEY .EQS. "OPTION_1" THEN ....
$ IF KEY .EQS. "OPTION_2" THEN ....
```

On NODE_B, when USER_B selects the option number 1 in the menu, it is automatically logged into NODE_A, with account USER_A and executes the application which is selected in the LOGIN.COM for option 1.

2. FastLogin Management

2.1 Declaring trusted remote DECnet users

FastLogin uses the standard DECnet proxy database. This database is managed using the AUTHORIZE utility. If a remote user is authorized for the traditional DECnet proxy access, it is automatically authorized for FastLogin.

FastLogin uses only the *default proxy* for each remote user. Consider the following proxy declaration on NODE_A:

```
UAF> ADD/PROXY NODE_B::USER_B USER_A/DEFAULT, USER_Z
```

When USER_B on NODE_B connects to NODE_A using a SET HOST command, its default user name will be USER_A. The non-default proxy for USER_Z is not used by FastLogin.

2.2 Enabling automatic remote DECnet login

For security reason, by default, FastLogin does not authorize any kind of automatic login. Each kind of remote source must be explicitely enabled by the system manager. Later in this documentation, we will see how each functionality can be enabled.

Thus, the automatic login for remote DECnet users, using the DECnet proxy database, is not enabled by default. You must define the logical name FASTLOGIN_ALLOW_DECNET_PROXIES in the system table to enable this functionality. This name must be defined in the system table, in executive mode:

```
$ DEFINE/SYSTEM/EXEC FASTLOGIN_ALLOW_DECNET_PROXIES TRUE
```

2.3 Declaring trusted remote LAT users

When a LAT connection is started from an OpenVMS node using the SET HOST/LAT command or a DECserver software emulator like DDE, the source of the connection is not identified with a DECnet node name and a username. Instead, the LAT node name and the UIC of the remote user are provided.

Since there is no standard proxy database for LAT, FastLogin uses the DECnet proxy database for incoming LAT connection also. FastLogin uses the ability to store a UIC in the proxy database to store *LAT proxies*. Of course, we can store a LAT proxy in the DECnet proxy database **only if the LAT node name of the source matches its DECnet node name**. However, on each system, the system manager usually assigns the same name to LAT and DECnet.

Example:

On NODE_B, the LAT node name is equal to the DECnet node name. On this sytem, the user named USER_B has UIC [20,3]. On NODE_A, we declare a proxy to USER_A using the following command:

```
UAF> ADD/PROXY NODE_B::[20,3] USER_A/DEFAULT
```

When USER_B (with UIC [20,3]) on NODE_B connects to NODE_A using a SET HOST/LAT command, its default user name will be USER_A.

2.4 Enabling automatic remote LAT login

Just like DECnet, the automatic login for remote LAT users, using the UICs in the DECnet proxy database, is not enabled by default. You must define the logical name FASTLOGIN_ALLOW_LAT_PROXIES in the system table to enable this functionality. This name must be defined in the system table, in executive mode:

```
$ DEFINE/SYSTEM/EXEC FASTLOGIN_ALLOW_LAT_PROXIES TRUE
```

Caution: On several versions or OpenVMS, at least up to versions 7.0, it has been observed intermittent failures of the SECURITY_SERVER process when the proxy database is searched for a non-existent proxy with a UIC. This is due to a bug in the OpenVMS security server. Consequently, unless the configuration has been fully tested, it is advised to disable FastLogin LAT proxies. On the other hand, there is no known problem with the DECnet proxies.

2.5 FastLogin inside a VMScluster

The whole community of users inside a VMScluster is assumed to share a single security domain. For this reason, FastLogin implements an automatic login facility inside each VMScluster:

- If USER_B starts a connection from NODE_B to NODE_A,
- if NODE_A contains no proxy for remote user NODE_B::USER_B (DECnet) or NODE_B::[uic] (LAT),
- if NODE_A and NODE_B are currently in the same VMScluster,

then the default user-name on NODE_A will be USER_B.

In other words, FastLogin can preserve the user-name across all SET HOST or SET HOST/LAT commands inside the same VMScluster.

As you can see in the above algorithm, with FastLogin the proxy database takes precedence over the implicit VMScluster login.

2.6 Enabling automatic login inside a VMScluster

Just like remote connections using proxies, the automatic login inside a VMScluster is not enabled by default. Again, FastLogin distinguishes two types of sources: DECnet and LAT. Both types may be individually enabled or disabled. This way, the system manager can decide which protocols he trusts on a security point of view.

To enable each type of automatic intra-cluster login, define the logical names FASTLOGIN_ALLOW_DECNET_CLUSTER and FASTLOGIN_ALLOW_LAT_CLUSTER in the system table. These names must be defined in the system table, in executive mode:

```
$ DEFINE/SYSTEM/EXEC FASTLOGIN_ALLOW_DECNET_CLUSTER TRUE
$ DEFINE/SYSTEM/EXEC FASTLOGIN_ALLOW_LAT_CLUSTER TRUE
```

For compatibility with the previous version of FastLogin, both types of connection can be enabled at once using the logical name FASTLOGIN_ALLOW_CLUSTER in the system table. This name must be defined in the system table, in executive mode:

```
$ DEFINE/SYSTEM/EXEC FASTLOGIN_ALLOW_CLUSTER TRUE
```

2.7 Managing a cluster alias

FastLogin recognizes the *cluster alias*. If NODE_A and NODE_B are inside a VMScluster and both nodes use a cluster alias CLUST1, the remote user who connects from NODE_B to NODE_A is seen as CLUST1::USER_B, not as NODE_B::USER_B.

On NODE_A, FastLogin recognizes that CLUST1 is its cluster alias and understands that the connection comes from the same VMScluster. Consequently, the implicit intra-cluster authentication applies.

This is the most common case. However, there may be cases were some nodes inside a VMScluster use a cluster alias when other nodes don't.

Back to the previous example, let's assume that NODE_B uses the cluster alias and NODE_A does not. In that case, the incoming connection which is identified by CLUST1::USER_B will not be recognized by NODE_A as coming from the same VMScluster since NODE_A has no cluster alias.

To handle this situation, the system manager can declare, on NODE_A, that some other nodes inside the same VMScluster use a cluster alias, although NODE_A does not. To do so, define on NODE_A the logical name FASTLOGIN_CLUSTER_ALIAS with the cluster alias. This name must be defined in the system table, in executive mode:

```
$ DEFINE/SYSTEM/EXEC FASTLOGIN_CLUSTER_ALIAS CLUST1
```

Note that this name will take precedence over another cluster alias on NODE_A.

2.8 Version restrictions

2.8.1 FastLogin restriction on OpenVMS versions before 6.1

FastLogin requires OpenVMS Alpha or VAX version 6.1 or higher.

FastLogin uses the *LOGINOUT callouts*. This feature was first introduced with OpenVMS V6.1. For this reason, FastLogin cannot be used on older VMS versions. Note that this requirement applies only to the node you want to *log into*. The node *from which* the connection is initiated can run any version of OpenVMS (or can even run any other operating system).

2.8.2 Restriction on OpenVMS Alpha version 6.1

OpenVMS Alpha version 6.1 provides no standard way to access the DECnet proxy database. For this reason, on this version, FastLogin does not use the proxy database. The only feature of FastLogin which can be used on OpenVMS Alpha version 6.1 is the automatic login inside a single VMScluster. If you want to use FastLogin with the proxy database on OpenVMS Alpha, upgrade to version 6.2 or higher.

There is no such restriction on OpenVMS VAX version 6.1.

3. Installation

FastLogin uses only one file: a shareable image named FASTLOGIN.EXE in the directory SYS$COMMON:[SYSLIB]. There is no installation procedure. This file must be installed by hand, using the following instructions.

FastLogin is only distributed in source form. There is only one source file called FASTLOGIN.C. This file must be compiled using the DEC C compiler, either on Alpha or VAX systems. Note that the obsolete VAX C compiler cannot be used since it does not understand some ANSI C constructs.

3.1 Rebuilding FastLogin

To rebuild FastLogin on OpenVMS Alpha systems:

```
$ CC FASTLOGIN.C
$ LINK/SHARE=FASTLOGIN.EXE FASTLOGIN.OBJ, SYS$INPUT:/OPTIONS
SYMBOL_VECTOR=(LGI$LOGINOUT_CALLOUTS=DATA)
$
```

To rebuild FastLogin on OpenVMS VAX systems:

```
$ CC FASTLOGIN.C
$ LINK/SHARE=FASTLOGIN.EXE FASTLOGIN.OBJ, SYS$INPUT:/OPTIONS
UNIVERSAL=LGI$LOGINOUT_CALLOUTS
$
```

3.2 Installing FastLogin

After rebuilding the FASTLOGIN.EXE image, copy it into SYS$COMMON:[SYSLIB].

```
$ COPY FASTLOGIN.EXE SYS$COMMON:[SYSLIB]
$ SET FILE SYS$COMMON:[SYSLIB]FASTLOGIN.EXE -
    /PROTECTION=(S:RWED,O:RWED,G:E,W:E) /OWNER=SYSTEM
```

Install the FASTLOGIN.EXE image using the INSTALL utility. Note that this is a requirement for shareable images containing LOGINOUT callouts. Also add this statement in the system startup file SYS$MANAGER:SYSTARTUP_VMS.COM:

```
$ INSTALL ADD SYS$SHARE:FASTLOGIN.EXE
```

Define the logical name LGI$LOGINOUT_CALLOUTS in the system table and add this definition in the system startup file SYS$MANAGER:SYSTARTUP_VMS.COM:

```
$ DEFINE/SYSTEM/EXEC LGI$LOGINOUT_CALLOUTS FASTLOGIN
```

Depending on the kind of automatic login you want to allow, define any of the following logical names in the system table and add these definitions in the system startup file SYS$MANAGER:SYSTARTUP_VMS.COM.

```
$ DEFINE/SYSTEM/EXEC FASTLOGIN_ALLOW_DECNET_PROXIES TRUE
$ DEFINE/SYSTEM/EXEC FASTLOGIN_ALLOW_LAT_PROXIES TRUE ! caution !
$ DEFINE/SYSTEM/EXEC FASTLOGIN_ALLOW_DECNET_CLUSTER TRUE
```

```
$ DEFINE/SYSTEM/EXEC FASTLOGIN_ALLOW_LAT_CLUSTER TRUE
```

Set the SYSGEN parameter LGI_CALLOUTS to 1. This parameter is dynamic and can be modified without rebooting the system:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET LGI_CALLOUTS 1
SYSGEN> WRITE ACTIVE
SYSGEN> EXIT
$
```

Also modify this parameter in the parameter file (SYSGEN> WRITE CURRENT) and in the SYS$SYSTEM:MODPARAMS.DAT text file.

4. FastLogin Troubleshooting

4.1 Tracing login problem

When logical name FASTLOGIN_TRACE is defined in the system logical name table and its translation is "TRUE", trace messages are displayed during login.

```
$ DEFINE/SYSTEM FASTLOGIN_TRACE TRUE
```

Example:

The following session shows a user who logs into another node in the same cluster using DECnet. Since there is no proxy for the remote user-name, the same name is used. FastLogin trace has been enabled on NODE_A.

```
$ SET HOST NODE_A
FastLogin facility is active
allow_decnet_cluster = 1
allow_lat_cluster    = 1
allow_decnet_proxies = 1
allow_lat_proxies    = 0

Identifying user
Job type is 5
Terminal flag is 1
Terminal device is "_RTA1:"
Terminal port is "NODE_B::USER_B"
Node is "NODE_B", user is "USER_B"
Looking for a proxy for "NODE_B::USER_B"
%SECSRV-E-NOSUCHPROXY, no proxy record matches your specification
Checking if "NODE_B" is inside the VMScluster
%SYSTEM-S-NORMAL, normal successful completion
Username [USER_B]:
User "USER_B" is trusted

   Welcome to OpenVMS Alpha Operating System, Version ...
    Last interactive login on ...
    Last non-interactive login on ...

$
```

Using LAT instead of DECnet, the session looks like:

```
$ SET HOST/LAT NODE_A
%LAT-S-CONNECTED, session to NODE_A established
%LAT-I-TODISCON, type ^\ to disconnect the session
FastLogin facility is active
allow_decnet_cluster = 1
allow_lat_cluster    = 1
allow_decnet_proxies = 1
allow_lat_proxies    = 0

Identifying user
Job type is 3
Terminal flag is 1
Terminal device is "_LTA2527"
Terminal port is "NODE_B/UIC_000020000003"
Node is "NODE_B", UIC group is 000020, member is 000003
Checking if "NODE_B" is inside the VMScluster
%SYSTEM-S-NORMAL, normal successful completion
Convert UIC [20,3] to username
%SYSTEM-S-NORMAL, normal successful completion
Username [USER_B]:
User "USER_B" is trusted

   Welcome to OpenVMS Alpha Operating System, Version ...
    Last interactive login on ...
    Last non-interactive login on ...

$
```

These examples also show how the remote user is seen. We can see that the node name is "NODE_B" and not a cluster alias or a decimal number (a DECnet address). In the later case, this would have indicated that the remote node is not declared in the local DECnet database or in the DNS namespace.

4.2 Sending FastLogin trace to a file or OPCOM

Using the logical name FASTLOGIN_OUTPUT_DEVICE in the system table, the trace message which were shown above may be redirected to a file:

```
$ DEFINE/SYSTEM FASTLOGIN_OUTPUT_DEVICE SYS$MANAGER:FASTLOGIN.LOG
```

If not device or directory is supplied, the file is created into SYS$MANAGER.

If the translation of this logical name is the special string «OPCOM», then the FastLogin output is sent as OPCOM messages of class security. To view them, type:

```
$ DEFINE/SYSTEM FASTLOGIN_OUTPUT_DEVICE OPCOM
$ REPLY/ENABLE=SECURITY
```

Later, when a remote user attempts a connection, you will see messages like:

```
$
%%%%%%%%%%  OPCOM  17-APR-1996 21:38:40.80  %%%%%%%%%%
Message from user <login> on NODE_A
Job type is 3
```

```
Terminal flag is 1
Terminal device is "_LTA2528"
Terminal port is "NODE_B/UIC_000020000003"

$
```

4.3 Disabling FastLogin without rebooting

If, for some reason, login into the system becomes impossible because of FastLogin, you should disable FastLogin. If a privileged user is already logged in, you do not need to reboot the system.

To disable FastLogin, issue the following commands:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET LGI_CALLOUTS 0
SYSGEN> WRITE ACTIVE
SYSGEN> EXIT
$ DEASSIGN/SYSTEM/EXEC LGI$LOGINOUT_CALLOUTS
```

4.4 Disabling FastLogin when rebooting

If no privileged user is already logged into the system and nobody can log in, the previous method cannot be applied. You have to halt the system. Then, perform a conversational boot. When you reach the SYSBOOT prompt, issue the following commands:

```
SYSBOOT> SET LGI_CALLOUTS 0
SYSBOOT> CONTINUE
```

Fortunately, such an extreme situation is not expected to happen.

4.5 Last resource...

In case of problem, send an E-mail to the author at:

```
lelegard@swthom.enet.dec.com
```