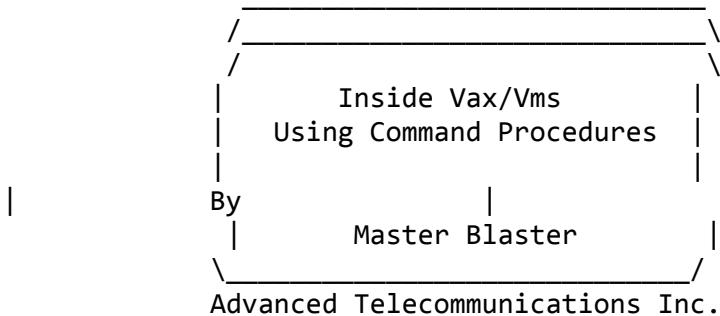


VAXVMS.TXT



Note: The following is geared for the more advanced hacker.

Part 1: Using Command Procedures.

You can use command procedures to automate sequences of commands that you use quite often. For example, if you always use the DIRECTORY command after you move to a Sub-Directory where work files are kept, you can write a simple command procedure to issue the SET DEFAULT and DIRECTORY commands for you. The following example, GO_DIR.COM, contains two commands:

```
$ Set Default [perry.accounts]  
$ directory
```

Instead of using each command alone, you can execute GO_DIR.COM with the @ command:

```
$ @Go_Dir
```

This command tells the DCL command interpreter to read the file GO_DIR.COM and executes the commands in the file. So the command interpreter sets your default directory to [PERRY.ACCTS] and issues the DIRECTORY command.

Note: DCL means Digital Command Language. (sorry)

Formatting Command Procedures

Use the DCL command CREATE to create and format a command procedure. When you name the command procedure, use the default file type COM. If you use this default, you don't have to use the file type when you execute the procedure with the @ command.

Command procedures contain DCL commands that you want the DCL command interpreter to execute and data lines that are used by these commands. Commands must begin with a dollar sign. You can start the command string just after the dollar sign.

Data lines do not start with a dollar sign. Data lines are used

VAXVMS.TXT

as input data for commands. Data lines are used by the most recently issued command.

The following example shows command and data lines in a command procedure.

```
$ mail
SEND
HOMAS
Y MEMO
id you get my memo?
```

Show users thomas

The first line is a command and must start with a "\$". The next lines are data lines that are used by the mail function; these lines must not start with "\$".

Using Multiple Lines for One Command

If you are writing a command that includes many qualifiers, you can make the command procedure more readable by listing the qualifiers on separate lines rather than running them together. To do this, use the hyphen as a continuation character. Don't start the continued line with "\$". For example:

```
$ print test.out -
    /after=18:00 -
    /copies=10 -
    /queue=lpb0:
```

Executing Command Procedures

You can execute command procedures in two modes: interactive and batch. In interactive mode, the commands in the command procedure are executing as if you were typing them. You cannot execute any other commands from your computer. In batch mode, the system creates a separate process to run the command procedure. After you use a batch job you can continue to use the system while it executes.

Executing Command Procedures Interactively

To execute a command procedure interactively, type the "@" command followed by the file specifications. If you don't enter the entire command specification, the system will use the current disk, directory, and file default.

Changing Command Levels

A command level is the DCL level from which you issue

VAXVMS.TXT

commands. When you log in and type commands at your Computer, you are issuing commands at your level zero. If you execute a procedure, the commands in the procedure are executed at command level 1. When the procedure ends and the DCL prompt is on your screen, you are back at level zero.

A System Login File

If a system login file exists, it is executed before the personal file. When the system login file ends, control is passed to the personal login file. System and group login files allow a system manager to make sure certain files are executed when a person logs in.

To make a system login file, you have to have a managers account, you use the name SYS\$SYLOGIN to make the login file.

Personal Login Files

After executing a system or group login file, the system executes a personal login file. Use a personal login file to execute commands that you want to that you want to issue everytime you login. Name the login command procedure LOGIN.COM and put it in the default login directory.

Defining Parameters or Qualifiers

You can create a command procedure that specifies only parameters and(or) qualifiers and then use the command procedure within a DCL command string. This type of command procedure is useful when there is a set of parameters or qualifiers that you frequently use with one or more particular commands. To execute the command string where you would normally use the qualifiers or parameters.

For example: You could create a command procedure that contains these qualifiers.

/DEBUG/SYMBOL_TABLE/MAP/FULL/CROSS_REFERENCE

To use this command procedure, execute it on the command line where you would otherwise place qualifiers. For example, if you name the command procedure DEFLINK.COM, you would use the following command line to link to an object module named YNAPSE.OBJ with the qualifiers that you specified in the command procedure:

\$ LINK SYNAPSE@DEFLINK

The next example shows a command procedure named PARM.COM that contains parameters:

HAP1, CHAP2

VAXVMS.TXT

To execute the procedure, use it in a command string in place of a parameter name:
\$ DIRECTORY @PARAM

s the others in this set are completed, they should stay in a
oup.

H)ackRite 1986 - Advanced Telecommunications Inc.

DOWNLOADED FROM P-80 SYSTEMS.....