Security: Why do I get error messages after patching BASH for the Shellshock vulnerabilty?

Why do I get error messages after patching BASH for the Shellshock vulnerabilty?

Symptom:

```
ash: BASH_FUNC_module(): line 0: syntax error near unexpected token `)
'
bash: BASH_FUNC_module(): line 0: `BASH_FUNC_module() () {  eval `/cm/
local/apps/environment-modules/3.2.6//Modules/$MODULE_VERSION/bin/modu
lecmd bash $*`'
bash: error importing function definition for `BASH_FUNC_module'
bash: BASH_FUNC_module(): line 0: syntax error near unexpected token `
)'
bash: BASH_FUNC_module(): line 0: `BASH_FUNC_module() () {  eval `/cm/
local/apps/environment-modules/3.2.6//Modules/$MODULE_VERSION/bin/modu
lecmd bash $*`'
bash: error importing function definition for `BASH_FUNC_module'
bash: BASH_FUNC_module(): line 0: syntax error near unexpected token `
)'
bash: BASH_FUNC_module(): line 0: `BASH_FUNC_module() () {  eval `/cm/
local/apps/environment-modules/3.2.6//Modules/$MODULE_VERSION/bin/modu
lecmd bash $*`'
```

The patching of BASH due to the shellshock vulnerability has introduced some changes with respect to the function definition syntax in exported functions

If you defined and export a function:

```
$ myfunction () { echo "hello world"; }
$ export -f myfunction
```

the corresponding environment variable is now different.

Security: Why do I get error messages after patching BASH for the Shellshock vulnerabilty?

The old form:

```
myfunction=() { echo "hello world"}
```

The new form:

```
BASH_FUNC_myfunction()=() { echo "hello world"}
```

The two forms are not compatible and this triggers the errors that you see.

The workaround is to source the modules init script again in your (job submission) script:

```
source /etc/profile.d/modules.sh
```

Unique solution ID: #1239
Author: Panos Labropoulos
Last update: 2014-10-30 16:54