NAME

loader_lua - kernel bootstrapping final stage

DESCRIPTION

The program called **loader_lua** is the final stage of FreeBSD's kernel bootstrapping process. On IA32 (i386) architectures, it is a *BTX* client. It is linked statically to libsa(3) and usually located in the directory */boot*.

It provides a scripting language that can be used to automate tasks, do pre-configuration or assist in recovery procedures. This scripting language is roughly divided in two main components. The smaller one is a set of commands designed for direct use by the casual user, called "builtin commands" for historical reasons. The main drive behind these commands is user-friendliness. The bigger component is the Lua interpreter.

During initialization, **loader_lua** probes for a console and sets the *console* variable, or sets it to serial console ("comconsole") if the previous boot stage used that. If multiple consoles are selected, they are listed separated by spaces. Then, devices are probed, *currdev* and *loaddev* are set, and *LINES* is set to 24. Next, Lua is initialized, and */boot/lua/loader.lua* is processed if it exists. After that, */boot/loader.conf* is processed if available.

At this point, if an **autoboot** has not been attempted, and if *autoboot_delay* is not set to "NO" (case insensitive), then an **autoboot** is attempted. If the system gets past this point, *prompt* is set and **loader_lua** enters interactive mode. Please note that, historically, even when *autoboot_delay* is set to "0", the user can interrupt the autoboot process by pressing a key on the console while the kernel and modules are being loaded. To prevent this set *autoboot_delay* to "-1". In this case **loader_lua** enters interactive mode only if **autoboot** has failed.

BUILTIN COMMANDS

In **loader_lua**, builtin commands take parameters from the command line. Presently, the only way to call them from a script is by using *evaluate* on a string. If an error condition occurs, an exception is generated, which can be intercepted using Lua exception handling. If not intercepted, an error message is displayed and the interpreter's state is reset, emptying the stack and restoring interpreting mode.

The commands are described in the loader_simp(8) "BUILTIN COMMANDS" section.

BUILTIN ENVIRONMENT VARIABLES

The environment variables common to all interpreters are described in the loader_simp(8) "BUILTIN ENVIRONMENT VARIABLES" section.

BUILTIN PARSER

When a builtin command is executed, the rest of the line is taken as arguments, and it is processed by a special parser which is not used for regular Lua commands.

SECURITY

Access to the **loader_lua** command line provides several ways of compromising system security, including, but not limited to:

- Booting from removable storage, by setting the *currdev* or *loaddev* variables
- Executing a binary of choice, by setting the *init_path* or *init_script* variables
- Overriding ACPI DSDT to inject arbitrary code into the ACPI subsystem

One can prevent unauthorized access to the **loader_lua** command line by setting the *password*, or setting *autoboot_delay* to -1. See loader.conf(5) for details. In order for this to be effective, one should also configure the firmware (BIOS or UEFI) to prevent booting from unauthorized devices.

MD

Memory disk (MD) can be used when the **loader_lua** was compiled with *MD_IMAGE_SIZE*. The size of the memory disk is determined by *MD_IMAGE_SIZE*. If MD available, a file system can be embedded into the **loader_lua** with */sys/tools/embed_mfs.sh*. Then, MD is probed and set to *currdev* during initialization.

Currently, MD is only supported in loader.efi(8).

FILES

/boot/loader	**loader_lua** itself.
/boot/defaults/loader.conf	
/boot/lua/loader.lua	Loader init
/boot/loader.conf	
/boot/loader.conf.local	**loader_lua** configuration files, as described in loader.conf(5).

EXAMPLES

Boot in single user mode:

```
boot -s
```

Load the kernel, a splash screen, and then autoboot in five seconds. Notice that a kernel must be loaded before any other **load** command is attempted.

```
load kernel
load splash_bmp
load -t splash_image_data /boot/chuckrulez.bmp
autoboot 5
```

Set the disk unit of the root device to 2, and then boot. This would be needed in a system with two IDE disks, with the second IDE disk hardwired to ada2 instead of ada1.

```
set root_disk_unit=2
boot /boot/kernel/kernel
```

Set the default device used for loading a kernel from a ZFS filesystem:

```
set currdev=zfs:tank/ROOT/knowngood:
```

ERRORS

The following values are thrown by **loader_lua**:

100	Any type of error in the processing of a builtin.
-1	**Abort** executed.
-2	**Abort"** executed.
-56	**Quit** executed.
-256	Out of interpreting text.
-257	Need more text to succeed -- will finish on next run.
-258	**Bye** executed.
-259	Unspecified error.

SEE ALSO

libsa(3), loader.conf(5), tuning(7), boot(8), btxld(8)

HISTORY

The **loader_lua** first appeared in FreeBSD 12.0.