Compile XigmaNAS loader.efi Yourself

This is the easiest way for building the loader.efi with increased EFI_STAGING_SIZE, it also assumes that the user has a bit of understanding of FreeBSD, including adding ports, packages and using any text editor.

Let's start prepare the FreeBSD development environment.

1. Install FreeBSD 13.2 RELEASE.
On your dedicated PC (or under a Virtualbox (on XigmaNAS system), VMware/Qemu) install and setup FreeBSD, it need less than 15GB total hard drive space for only building the loader.efi, also setup the networking services as they are required to download the required source files and packages, then reboot to complete system install.

You now can login as root.

2. Update FreeBSD.
Now, update your installed copy of FreeBSD with the latest patches.

2.1. Security update.
Begin with installing the latest security patches:
freebsd-update fetch install

2.2. Now reboot the system and login as root.
Root login is not necessary but really recommended. It's only the development environment.

2.3. Install the required ports.
Go into the system ports and make install clean.
cd /usr/ports/security/ca_root_nss && make install clean
cd /usr/ports/devel/git && make install clean

3. Create the working directory and CD to it.
This is the place where all source files will be stored.
mkdir /xigmanas
cd /xigmanas

3.1. Download the required FreeBSD sources.
git clone -b releng/13.2 https://git.FreeBSD.org/src.git

3.2. CD to the release sources directory.
cd src

3.3. Prepare the build environment.
make kernel-toolchain TARGET=amd64
make _includes TARGET=amd64
make buildenv TARGET=amd64 EFI_STAGING_SIZE=256

4. Now is time to compile and build the custom loader.efi.
You can ignore some compile errors at this first "stand" stage only*.
make -C stand all

4.1. Run make again on "stand/efi" to just compile the efi files we want, until the previous errors are sorted out*.
make -C stand/efi all

4.2. Copy the loader.efi files to the working directory.
cp /usr/obj/xigmanas/src/amd64.amd64/stand/efi/loader_4th/loader_4th.efi /xigmanas
cp /usr/obj/xigmanas/src/amd64.amd64/stand/efi/loader_lua/loader_lua.efi /xigmanas
cp /usr/obj/xigmanas/src/amd64.amd64/stand/efi/loader_simp/loader_simp.efi /xigmanas
cp /usr/obj/xigmanas/src/amd64.amd64/stand/efi/loader_lua/loader_lua.efi /xigmanas/loader.efi

5. Create an UEFI bootable image for the XigmaNAS ISO.
Copy and paste the below commands one by one, it should work.
cd /xigmanas
mkdir -p /mnt/efiboot
dd if=/dev/zero of=efiboot.img bs=9k count=100
md=$(mdconfig -a -t vnode -f efiboot.img)
newfs_msdos -F 12 -m 0xf8 ${md} 2>/dev/null
mount -t msdosfs /dev/${md} /mnt/efiboot
mkdir -p /mnt/efiboot/efi/boot
cp loader.efi /mnt/efiboot/efi/boot/bootx64.efi
echo BOOTx64.efi > /mnt/efiboot/efi/boot/startup.nsh
umount /mnt/efiboot && rm -r /mnt/efiboot
mdconfig -d -u ${md}

Note: "13.2" release was used in this guide to download and to work with, make sure the wanted platform and version match the system as well as for XigmaNAS target working platform. Good luck, now you can do it yourself!