

nas-tweaks.net

HDD-Installation of the fun_plug 0.7 on NAS-devices – NAS-Tweaks.net

Published by Uli Uli @ Google+, Facebook, Twitter. View all posts by Uli

The Firmwares of various NAS-Devices includes a very interesting bonus: the user can execute a script (file) named “*fun_plug*” when the OS is booted. Unlike all the other Linux software which is loaded when the NAS boots, this file is located on Volume_1 of the hard disk rather than within the flash memory. This means the user can easily and safely modify the file because the contents of the flash memory is not changed. If you delete the fun_plug file (see [here](#) for instructions), or replace your hard disk, the modification is gone.

Fun_plug allows the user to start additional programs and tools on the NAS. A Berlin-based developer named “Fonz” created a package called “f fp” (Fonz fun_plug), which includes the script and some extra software which can be invoked by fun_plug.

Installation of fun_plug is easy and takes only a few steps. These steps should be performed carefully, as they depend on typed commands and running with “root” privileges.

Contents

- [Purpose, risks, and benefits](#)
 - [Responsibility](#)
 - [Benefits](#)
 - [Technical synopsis](#)
- [Steps for installing fun_plug](#)
 - [Download](#)
 - [Option: view the fun_plug script](#)
 - [Reboot](#)
 - [Option: view ffp.log](#)
 - [Connect via telnet](#)
 - [Change root password](#)
 - [Activate SSH](#)
 - [Logging in using SSH](#)
 - [Now what?](#)

- [Notes](#)
 - [Fun_plug and user accounts](#)

Purpose, risks, and benefits

Fun_plug is essentially a technique to stepwise turn a NAS with fixed out-of-the-box functionality into an open Linux machine on which you can install additional software packages and, if you want, learn a bit about Linux.

Responsibility

This also implies that you are (temporarily or permanently) turning a stable turnkey system into a system that the respective vendors no longer supports. This is similar to buying a notebook with Microsoft software, and installing Linux on it. The shop where you bought it can no longer help you if you claim the audio no longer works.

Although there is a [Tutorial](#) on how to disable and even remove fun_plug, and although the authors have tested their recipes, checked the wording and added warnings, these are advanced tools which can, if you experiment more than your own know-how can handle, give advanced problems.

Risks involved in all this are not so much damaging your hardware (shouldn't be possible), but loss of reliability of the NAS (you bought a file server to reliably store files, didn't you). This risk may be acceptable because the software was preintegrated and tested by competent people. But you yourself are, at the end of the day,

responsible for deciding to use this.

Possibly a less obvious, but more real risk is that some kind of extensions to the NAS (e.g. adding a server) imply that you may decide to open your local network a bit to the outside world. For example, to allow others to view your holiday videos stored on the device. The out-of-the-box NAS can already have this problem (via the ftp server). The point here is that **you** are responsible for the security of your device and entire network. This site doesn't even have tutorials on basic security issues like firewalls, etc. because these are all NAS independent and the tutorials would never be foolproof anyway. So when used wrongly, the NAS and firewall obviously do allow others to read more data than you intended. Or to delete your valuable data. Or to replace software by other software (chance is small, but the impact is high).

Conclusion: as the NAS is a powerful networked device, and as these tutorials can help you make it even more powerful, you are responsibility for having the basic understanding of networked security. Again, this also applies to an out-of-the-box NAS. But the more you mess with it, the more you need to apply some common sense. This is incidentally the reason why we provide some explanation on what you are doing in the tutorials, rather than just telling you what to type



Benefits

The main reason why people go this route is to extend their NAS with servers such as [BitTorrent](#) clients or Web servers. Other typical uses are to add extensions which fix current limitations of the device (e.g. time accuracy, fan noise).

Technical synopsis

In a first step, we install a script named `fun_plug` that provides a hook to extend the [boot](#) process of Linux on the NAS. That hook was intentionally added by the vendor to enable this. But the vendors do not document or support all of this.

An initial set of packages (downloaded as a single compressed archive) gives you enough tools to get started and, if you are curious about the machine or its software, to carefully look around.

This set of tools gives you the ability to install even more software packages (typically servers) from trusted sources. These packages should obviously all have been compiled for the processor in the NAS and should have been tested on the device (or a very similar device) by a software expert.

Steps for installing fun_plug

Preparation

The NAS needs to have a valid network configuration. Check the

gateway & dns-servers to be valid. If the NAS receives the network configuration via DHCP it should be correct in most cases. If you receive the error “wget:bad address `wolf-u.li`” (or similar / other domains) the configuration needs to be checked.

Please ensure that the internal “Remote Backup” Services is deactivated. This server is shipped with newer NAS like the DNS-320/DNS-320L/DNS-325/DNS-345 and conflicts with the SSH-Server of the fun_plug. If you want to use the Backup-Service, please take a look at [this tutorial](#) after you have executed the installation of the fun_plug.

Determination of the correct fun_plug

Fonz has decided to build to version of the fun_plug for different devices. The Types are “EABI” and “OABI” which you can determine by clicking on the [description of your device here](#). Then search for “Application binary interface” in the details.

ATTENTION: The release notes of the D-Link DNS-320L Firmware 1.06 show: “Fun_plug script was removed according to security issue and no longer to be supported.” Therefore this version is NOT supported, please downgrade if you want to use ffp.

Download

Download the latest files:

- [fun_plug](#)
- fun_plug.tgz
 - [Version “ARM” for EABI-devices](#) (like the DNS-320/325/345 and CH3MNAS) ([Mirror](#))
 - [Version “OABI” for OABI-devices](#) (like the DNS-323 and CH3SNAS) ([Mirror](#))

Ensure that these two files are named fun_plug (not fun_plug.sh) and fun_plug.tgz (not fun_plug.gz or fun_plug.tar.gz or fun_plug.tar).

Place both files in the topmost directory of Volume_1 of your NAS. If you use Windows Explorer it could look like [this](#) or [this](#) or similar. If you are running a different OS like Linux or MAC OSX, please ensure that the file fun_plug is marked as executable (chmod 777).

Option: view the fun_plug script

For fun, you may want to open the file fun_plug. If your are on Windows choose [Notepad++](#) or any other “better” Editor. Avoid using Windows’ Notepad for viewing/editing Linux text files: Windows and Linux use different end-of-line conventions. Please be careful not to accidentally modify it.

The script fun_plug is an ASCII file with commands which are

executed by the Linux command interpreter (sh for “shell”).

Lines starting with “#” are comments (“#!/bin/sh” is a special case).

You might be able to decode that the program creates a log file called `ffp.log` (an ASCII file used here to capture the lines which start with “echo”).

Firstly, a number of named constants are defined for various file names and fragments of file names (the lines like “FFP_SOMETHING= . . .”).

You can see that Fonz developed it for a D-Link DNS-323 (rather than a Conceptronic [CH3SNAS](#), but this doesn’t matter as [Uli](#), [PeterH](#) and others have tested in on the CH3SNAS).

The command `date` will copy the current date and time to the log file.

Next, a first script `setup.sh` is run if it is found in the expected `/mnt/HD_a2/.bootstrap/` folder. Initially it will not be found.

Then a new directory “ffp” is created (`mkdir`) and the `fun_plugin.tar.gz` file is unpacked (`tar`) into that directory. This step is a bit more complex than normal due to a problem with the `tar` version supplied with the NAS. As a workaround `tar` is run twice (first the older version, and then the `tar` version which was untarred

from fun_plugin.tgz).

If all went well, the log file gets an extra “OK” string. And the tarball input file is deleted (rm). This obviously only happens once (the script skips the unpacking if the tarball file is not found using the `if [condition]; commands fi` construct).

The “chown” is about changing ownership for a program called [busybox](#). And “chmod” is about changing access privileges.

Then, a script file `/ffp/etc/fun_plugin.init` (“containing the ffp-scripts package”) is executed if it is detected.

Next, a script file `/ffp/etc/fun_plugin.local` is executed if it is detected. It can be used to add your own startup commands: it will not be overwritten by package updates.

Finally, a script file `/ffp/etc/rc` is run if it exists.

Reboot

Reboot the NAS by holding down the power button 5 seconds or via the web interface (Mostly “Tools” -> “System” -> “Reboot” or similar somewhere in the menu of the webinterface). This causes the NAS to go and find the file fun_plugin on Volume_1 and execute it.

Option: view ffp.log

If you are interested, you will find that the `fun_plug.tgz` tarball has disappeared, and has been unpacked into the newly created `ffp` directory.

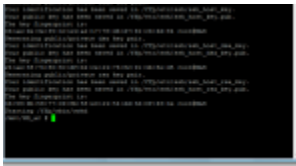
You will also find the `ffp.log` file created during execution of the `fun_plug` script and while executing some of its commands. It is longish (e.g. 47 KBytes) because the `tar` program generates a lot of warnings about repairing links (this only happens once). You can view the log file with WordPad or NotePad++.

From now on, whenever the NAS is rebooted and thus the `fun_plug` script is re-executed, the script appends about 15 extra text lines to the end of this log file. These contain the date/time of reboot and the status of various servers which you may enable in the future (see below). This appending of information to `ffp.log` gives you one way to determine whether `fun_plug` is really running: if you last reboot of the NAS is listed, `fun_plug` and any servers that it activates are running.

Note that the end of the initial log file already states that a server called `telnetd` is already running. We will use Telnet in the next step.

Connect via telnet





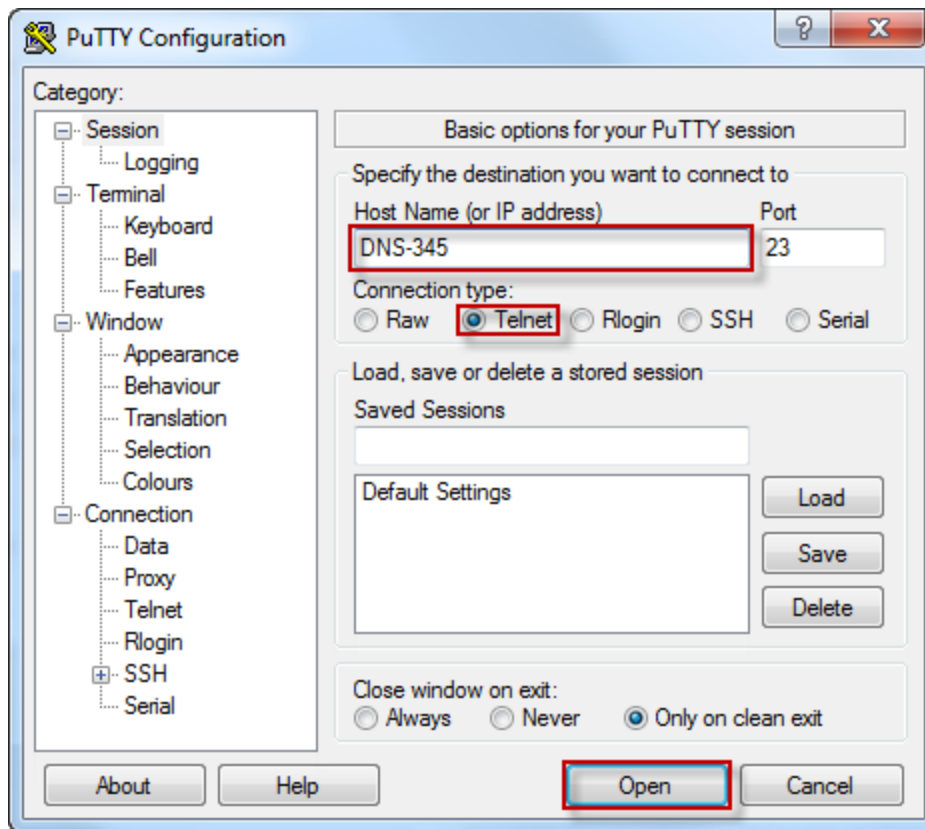
Telnet Session

After rebooting, you need to connect to the NAS using a protocol called [Telnet](#). Telnet allows you to “login” on a remote machine via a command line window.

Windows users can use an open-source telnet client called [PuTTY](#). PuTTY is a self-contained program: the PuTTY.exe file can be stored wherever convenient and executed without any prior installation. In the PuTTY configuration screen you need to set the following before pressing Open:

- Host name (or IP address): use the name of the share (e.g. CH3SNAS) or its IP address
- Select Connection type “Telnet” (which defaults to port 23)

Now you can press Open (PuTTY can save these settings under a default or name if you want, but you will likely be using ssh instead of telnet later on). In PuTTY these setting could look like this:



Linux users are “supposed to be” familiar with how to use telnet.

After connecting to the device, the first line telnet will show:

Or

If this doesn't show up, type “5784468” to get this prompt.

```
sh-4.1#
```

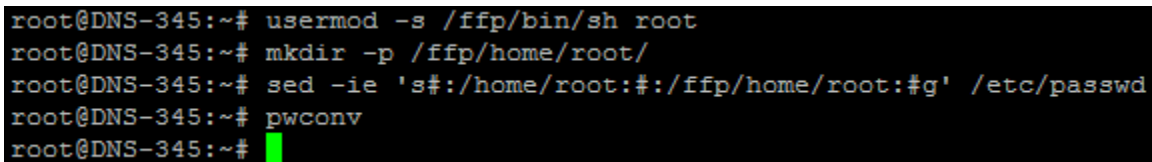
If you can see the prompt, you are logged in. This command “prompt” is where you can type in commands. The prompt shows you are in the root directory. Note that Linux command lines are not very communicative. These Rambo-like social skills are generally attributed to Linux' resource-deprived childhood.

Change root password

We proceed with updating /etc/shadow by using the program pwconv. It uses /etc/passwd to generate the necessary lines in the shadow-file. Also the default shell will be changed. Additionally the home-directory of root is changed to a permanent one:

```
usermod -s /ffp/bin/sh root
mkdir -p /ffp/home/root/
sed -ie 's#:/home/root:#:/ffp/home/root:#g' /etc/passwd
pwconv
```

This could look like this:



```
root@DNS-345:~# usermod -s /ffp/bin/sh root
root@DNS-345:~# mkdir -p /ffp/home/root/
root@DNS-345:~# sed -ie 's#:/home/root:#:/ffp/home/root:#g' /etc/passwd
root@DNS-345:~# pwconv
root@DNS-345:~#
```

Hints:

- If the following error comes up when using “usermod”:
"usermod: no changes", just ignore it. In this case the shell is already set to the correct one.
- If the following error comes up when using “pwconv”:
"pwconv: failed to change the mode of /etc/passwd- to 0600" just ignore it and proceed.

Now we need to change the password of user “root” to prevent unauthorized access.

Run the passwd command and enter a new password twice (note that Linux passwords are case-sensitive):

Now check if everything went right using:

This could look like:

```
root@DNS-345:~# passwd
Changing password for root
Enter the new password (minimum of 5 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
passwd: password changed.
root@DNS-345:~# login

DNS-345 login: root
Password:
No mail.
root@DNS-345:~#
```

If this was successful, proceed to the next step, otherwise return to “passwd”.

Store the password in the NAS. This step is essential, otherwise your password will be cleared on the next reboot! Also when you change the password of any user using the commandline, run the following `store-passwd.sh` again!

```
wget http://wolf-u.li/u/172/ -O /ffp/sbin/store-
passwd.sh
store-passwd.sh
```

This could look like this:

```
sh-4.1# wget http://wolf-u.li/u/172/ -O /ffp/sbin/store-passwd.sh
Connecting to wolf-u.li|83.169.42.106|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 875 [application/octet-stream]
Saving to: `/ffp/sbin/store-passwd.sh'

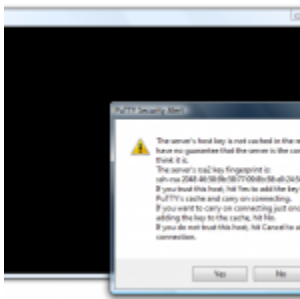
100%[=====>] 875          --.-K/s   in 0s

2012-02-21 19:29:49 (27.2 MB/s) - `/ffp/sbin/store-passwd.sh' saved
sh-4.1# store-passwd.sh
Saving Userdata to /usr/local/config/
sh-4.1#
```

Activate SSH

Now activate SSH (secure shell: telnet has major security limitations). Such lines can best be copied line-by-line or together into PuTTY:

```
chmod a+x /ffp/start/sshd.sh
sh /ffp/start/sshd.sh start
```



First Connection with SSH

Note that executing `sshd.sh` takes a while to execute and generates three pairs encryption keys for secure communication between the CH3SNAS and a remote client (computer). Each pair has a “fingerprint” for the public key and a corresponding graphical

“randomart” image. The fingerprint for the RSA encryption algorithm will incidentally show up again in the next step.

As shown in one of the pictures, the first time you connect to this new (as far as ssh is concerned) machine, you will get a stern warning from ssh. This is because ssh expects to be connecting to this machine through an encrypted connection (now and likely in the future). But ssh wants to be sure that you are connecting to the intended machine rather than to an imposter (“man-in-the-middle”) and has no way of knowing if this is the case. Assuming that you are connecting to via your own (safe) LAN, you don’t need to worry whether the presented identification (public-key fingerprint) is the right one. If you need to connect over the internet (very unlikely) or are paranoid (unlikely), you can follow the confirmation procedure described in [this website](#).

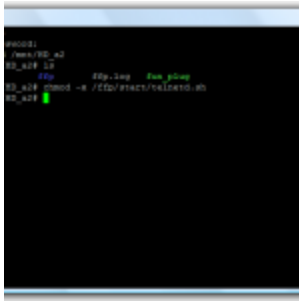
Note that this step associates the name and IP number of your NAS with this public key (this is stored on your computer). This means that during future ssh sessions to this machine the confirmation of the public key is done automatically.

Logging in using SSH

Now you can try to login using an ssh session as user root. This involves starting a second copy of PuttY.

Once you were logged in successfully, you can deactivate telnet using:


```
chmod -x /ffp/start/telnetd.sh
```



SSH Session

If the login was not successful, please check that you executed all necessary steps from above. If you still cannot login, please contact us below in the comments.

Note that at this point telnet is actually still running, but it will stop working the next time you reboot the NAS (DO NOT REBOOT YET!). Once you have tested that the ssh server and the associated root password, and encryption keys are working fine you can turn telnet off.

```
sh /ffp/start/telnetd.sh stop
```

Now what?

Congratulations! With the last step, you've installed your fun_plug

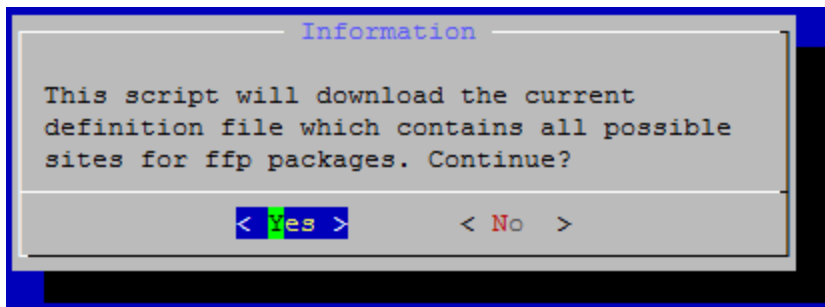


Now you need to execute a few steps to be able to install more packages.

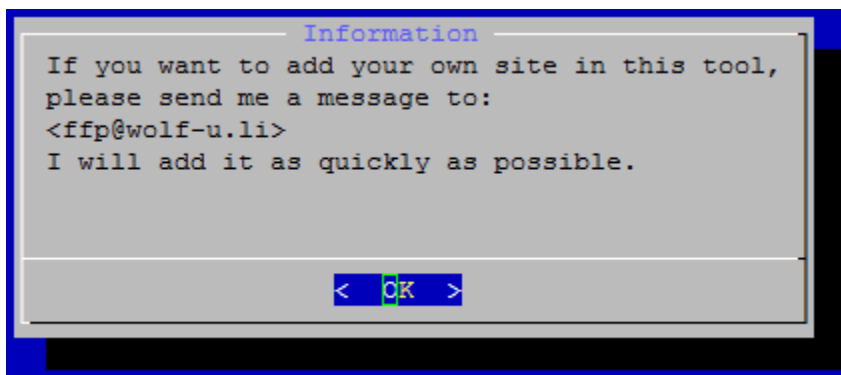
First the sources for packages are installed:

```
wget http://wolf-u.li/u/441 -O /ffp/bin  
/uwsiteloader.sh  
chmod a+x /ffp/bin/uwsiteloader.sh  
uwsiteloader.sh
```

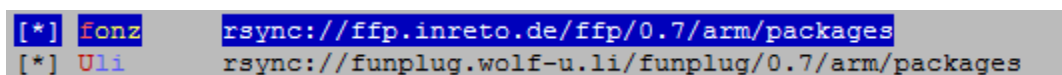
After entering the last line a menu pops up:



Press "Enter"

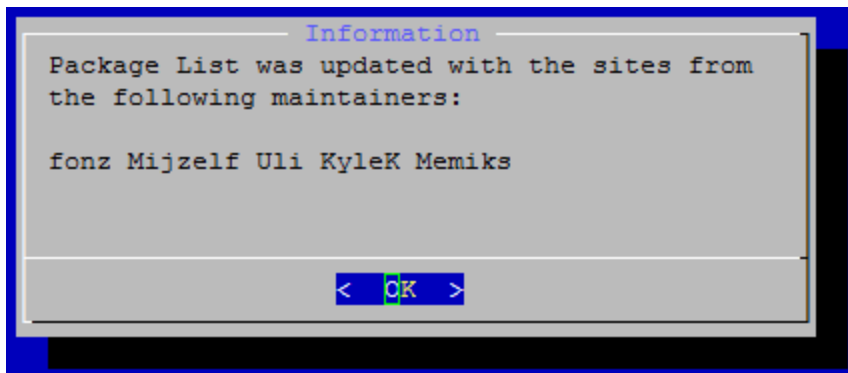


Press "Enter". Now a screen pops up where you can choose the sites.

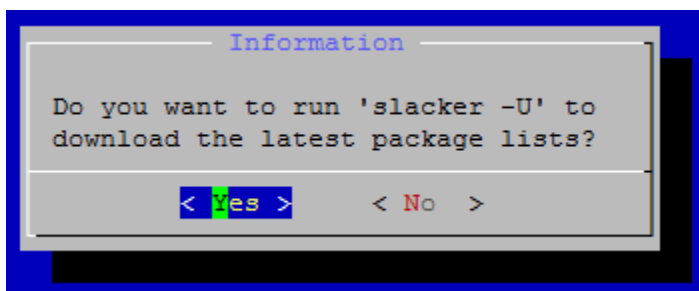


To choose an entry, you need to mark it with the arrow-keys on your keyboard and then press spacebar to activate it. It is activated if there is [*] in the first characters of this line. Please choose at

least “fonz” and “Uli”, otherwise the tutorials on this website will not work. At the end choose “OK” and press “Enter”.



Press “Enter”



Press “Enter”. This executed the update of the sites to see the current packages. This will not execute a installation. Now you need to install one more package before executing your next reboot:

To see what this package is about, you can go [here](#).

You can now [follow additional tutorials](#) or (carefully) look around using the [command line](#)!

Notes

Fun_plug and user accounts

Note that the initial execution of the fun_plug script creates a new user group utmp.

The script that installs the ssh server creates a new user named sshd and adds the user to utmp. This user is for internal use only, and has no ability to login. It is standard procedure when installing OpenSSH, and believed to be safe.

On a NAS, user sshd also shows up as having read-only ftp access to Volume_1. Although it is doubtful that this user really can access ftp, this seems to be a bug and is being investigated. er group utmp.

The script that installs the ssh server creates a new user named sshd and adds the user to utmp. This user is for internal use only, and has no ability to login. It is standard procedure when installing OpenSSH, and believed to be safe.

On a NAS, user sshd also shows up as having read-only ftp access to Volume_1. Although it is doubtful that this user really can access ftp, this seems to be a bug and is being investigated.