

How To Jailbreak A PogoPlug.....	3
Step One: Get Access	3
Step Two: Set Up Your Flash Drive.....	3
Step 3: Download and install Arch Linux ARM.....	4
Step 4: Set a Static IP Address & Set the Host Name.....	4
Step 5: Localization	4
Installation	6
Revert to Original PogoPlug FW.....	7
Installing anything with Pacman	8
Install Apache.....	9
How to Find your Server's IP address	10
Configure Apache.....	11
Install Curlftps.....	12
Install VSFTP FTP Service.....	12
Enabling uploading.....	12
Local user login.....	12
Enabling A Service At Startup	13
Set Static IP Address:	13
To install samba	14
To install Subversion (SVN)	14
Mount External Drive.....	15
YaBB Requirements.....	15
Install perl-net-smtp-ssl	16
Virtual Hosts.....	17
Example httpd-vhosts.conf file	17

Discussion Board	19
To Copy YaBB	19
Postfix (for mail).....	19
Useful File Locations	20
Useful Commands	20

How To Jailbreak A PogoPlug

Step One: Get Access

1. Plug the flash drive to the top USB socket (replace cover if desired).
2. Plug it in.
3. Plug in the network cable. Plug in the other end to an internet connection.
4. Wait for the LED to turn a constant green.
5. From a browser, navigate to my.pogoplug.com.
6. NEXT
7. NEXT
8. NEXT (if all is well)
9. Enter the ID (look under the Pogoplug) in the place provided.
10. NEXT
11. Click REGISTER NOW
12. Select the 5GB Cloud (free!) option
13. Skip the tour
14. Click SETTINGS on the far upper-right
15. Select SECURITY under Account Settings.
16. Check "Enable SSH access for this Pogoplug device. Note: if this does not appear, try refreshing, try cycling power. It does not always appear immediately.
17. Select a password for your SSH and enter it twice to confirm.

Step Two: Set Up Your Flash Drive

(source: archlinuxarm.org)

1. Find the local IP address of your Pogoplug: log into your router and look at the DHCP client table (usually under "status").
2. Using Putty or similar, connect to the Pogoplug using this local IP address, on port 22, SSH. User "root" with the password you established in step 18, above.
3. Enter "**killall hbwd**" to stop the Pogoplug software on the device.
4. Start fdisk to partition the USB2 or SATA drive:
/sbin/fdisk /dev/sda
 - 4.1. At the fdisk prompt, delete old partitions and create a new one:
 - 4.2. Type **o**. This will clear out any partitions on the drive.
 - 4.3. Type **p** to list partitions. There should be no partitions left.
 - 4.4. Now type **n**, then **p** for primary, **1** for the first partition on the drive, and then press ENTER, accepting default values.
 - 4.5. Write changes to disk and exit by typing **w**.
5. Create the ext3 file system and mount it:
cd /tmp
wget http://archlinuxarm.org/os/pogoplug/mke2fs
chmod +x mke2fs
./mke2fs -j /dev/sda1
mkdir alarm

```
mount /dev/sda1 alarm
```

Step 3: Download and install Arch Linux ARM

1. Get it:

```
cd alarm  
wget http://archlinuxarm.org/os/ArchLinuxARM-armv5te-latest.tar.gz  
wget http://archlinuxarm.org/os/ArchLinuxARM-kirkwood-latest.tar.gz
```
2. Install it:

```
tar -xvzf ArchLinuxARM-armv5te-latest.tar.gz  
tar -xzf ArchLinuxARM-kirkwood-latest.tar.gz
```

(may take a while)
3. Clean up and make sure all changes are written to the drive:

```
rm ArchLinuxARM-armv5te-latest.tar.gz  
rm -rf ArchLinuxARM-kirkwood-latest.tar.gz  
sync
```
4. Unmount the drive:

```
cd ..  
umount alarm
```
5. Download the U-Boot installer and run it:

```
cd /tmp  
wget http://archlinuxarm.org/os/ppv4/ppv4-install.sh  
chmod +x ppv4-install.sh  
./ppv4-install.sh
```

Step 4: Set a Static IP Address & Set the Host Name

Setting a static IP address will make sure your local IP remains the same, regardless of what other computing devices are attached to the local network. This becomes important when you configure your router to forward port 80 (http) to your Pogoplug. Setting the host name is optional, but it identifies the machine you're on at the terminal prompt, and also affects the name listed in the header of emails sent out.

1. Set the host name:

```
hostnamectl set-hostname <pick-a-host-name>
```
2. Set a static IP address:

```
ip addr add 192.168.1.222/24 broadcast 192.168.1.255 dev eth0 (assuming address is to be .222)  
ip route add default via 192.168.1.1
```

Better method is to reserve a fixe IP address on the router, letting the Pogoplug come up and request it as a dynamic address. This avoids having to set up any fixed address on the Pogoplug.

Step 5: Localization

Set locale

Modify /etc/local.gen. Uncomment line "en_US.UTF-8 UTF-8".

"Generate" by executing "locale-gen". I assume this propagates though the system somehow.

Create the file "locale.conf" with this setting as well, using: "echo LANG=en_US.UTF-8 > /etc/locale.conf".

Now "export" this by executing "export LANG=en_US.UTF-8"

Set the time zone.

Get available time zones for the us by typing "ls /usr/share/zoneinfo/US"

Create a "symbolic link" to the desired subzone:

```
"ln -s /usr/share/zoneinfo/US/New_York /etc/localtime"
```

If it fails because "file exist", check the current setting with:

```
"ls -l /etc/localtime", and overwrite it as necessary with by adding the -f option to the ln command above: "ln -f -s /usr/share/zoneinfo/US/New_York /etc/localtime"
```

AND/OR

Get timezones using "timedatectl list-timezones"

And set using "timedatectl set-timezone America/New_York" this seems to set it so the time shows correctly when using the "date" command, for instance.

TODO: find out if we need to set up automatic update from a network source.

Installation

1. With the device on and online, register and enable SSH through my.pogoplug.com.
2. Power down the original, unmodified Pogoplug Series 4.
3. With only the drive you intend to install Arch Linux ARM to plugged in (all data will be erased), switch on the power.
4. Log in to the Pogoplug Series 4 over SSH.
5. Stop the Pogoplug software so it doesn't interfere with the install process:

```
killall hbwd
```

6. Start fdisk to partition the USB2 or SATA drive:

```
/sbin/fdisk /dev/sda
```
7. At the fdisk prompt, delete old partitions and create a new one:
 - a. Type **o**. This will clear out any partitions on the drive.
 - b. Type **p** to list partitions. There should be no partitions left.
 - c. Now type **n**, then **p** for primary, **1** for the first partition on the drive, and then press ENTER, accepting default values.
 - d. Exit by typing **w**.

Create the ext3 filesystem:

```
cd /tmp
```

```
wget http://archlinuxarm.org/os/pogoplug/mke2fs
```

```
chmod +x mke2fs
```

```
./mke2fs -j /dev/sda1
```

```
mkdir alarm
```

```
mount /dev/sda1 alarm
```

Download and install Arch Linux ARM:

```
cd alarm
```

```
wget http://archlinuxarm.org/os/ArchLinuxARM-armv5te-latest.tar.gz
```

```
tar -xvzf ArchLinuxARM-armv5te-latest.tar.gz
```

```
# This will take a long time
```

```
rm ArchLinuxARM-armv5te-latest.tar.gz
```

```
sync
```

```
# Takes a while if you are using a flash drive
```

Unmount the drive:

```
cd ..
```

```
umount alarm
```

Download the U-Boot installer and run it:

```
cd /tmp
```

```
wget http://archlinuxarm.org/os/ppv4/ppv4-install.sh
```

```
chmod +x ppv4-install.sh
```

```
./ppv4-install.sh
```

8. After the installer tells you it's done, and if there were no errors, reboot:

```
/sbin/reboot
```

Double-check your router's DHCP tables to see if a different IP was leased, and you can now SSH in with the user/pass of root/root.

- Also: changed root password after reboot:

Passwd

Revert to Original PogoPlug FW

Simply create a file called "revert" on a flash drive, insert in the top USB slot and re-power the PogoPlug.

Installing anything with Pacman

First run:

Pacman -Syu

To make sure pacman itself and database is up-to-date.

Install Apache

Apache is a free open source software that runs over 50% of the world's web servers.

Note: as root, you do not need to prefix "sudo" with these commands.

Before installing any of the LAMP programs, we should update the package manager:

```
sudo pacman -Syu
```

Once the update is complete, we can install Apache:

```
sudo pacman -S apache
```

After Apache is installed, we need to make a couple of changes in the configuration.

Open up the apache configuration file *note: I use "vi" not "nano" to edit:*

```
sudo nano /etc/httpd/conf/httpd.conf
```

Comment out the `unique_id_module` (you can use `ctrl w` to find it quickly):

```
#LoadModule unique_id_module modules/mod_unique_id.so
```

Restart Apache:

```
sudo systemctl restart httpd
```

During the apache restart, you may get a notice that says the following:

```
httpd: apr_sockaddr_info_get() failed for droplet1
httpd: Could not reliably determine the server's fully qualified domain name, using
127.0.0.1 for ServerName
```

[DONE]

Although this alert does not interfere with the apache launch, you can easily eliminate it if you like by adding your hostname to your configuration.

Open up the hosts file:

```
sudo nano /etc/hosts
```

Add your hostname to the end of the line beginning with 127.0.0.1:

```
127.0.0.1    localhost.localdomain localhost droplet1
```

Subsequent restarts of Apache will no longer display that message.

With that, Apache is now installed on your server. Directing your browser to your server's IP address (<http://12.34.56.789>) will display an auto-index directory.

You can quickly try out a sample page by adding an index.html file to Arch's document root directory, located within "srv/http":

```
sudo nano /srv/http/index.html

<html>
<title>Welcome</title>
<body>
<h2>Hello, Welcome to Arch</h2>
</body>
</html>
```

You can visit the placeholder page by going to your server's IP address in your browser.

How to Find your Server's IP address

You can run the following command to reveal your server's IP address.

```
curl -s icanhazip.com
```

Configure Apache

Uncomment/change the following lines in `/etc/httpd/conf/httpd.conf`:

```
#AddType text/html .shtml
#AddOutputFilter INCLUDES .shtml
```

In the “<Directory “/srv/http”>” section, add:
Options +Includes

```
<IfModule dir_module>
    DirectoryIndex index.shtml index.htm
</IfModule>
```

Near the other “AddHandler” lines, add:
AddHandler server-parsed .shtml

Add:

```
Redirect /index.htm http://ourcarpentervillage.com/index.shtml
Redirect /index.html http://ourcarpentervillage.com/index.shtml
Redirect /board http://www.ourcarpentervillage.com/cgi-bin/cvbb/YaBB.pl
Redirect /discussion http://www.ourcarpentervillage.com/cgi-bin/cvbb/YaBB.pl
Redirect /cvbb http://www.ourcarpentervillage.com/cgi-bin/cvbb/YaBB.pl
```

(note: need to rename the `/board` subdirectory to `/BOD`, to avoid getting redirected to the discussion board)

Restart Apache with:
systemctl restart httpd

Start Apache on boot:
systemctl enable httpd

Enable CGI-BIN:

Add line:
ScriptAlias /cgi-bin/ "/srv/cgi-bin/"

Also change to do the `/srv/cgi-bin` directory:

```
<Directory "/srv/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

From the command line, run:
`mkdir /srv/cgi-bin`

Add cgi files (Perl, etc.) as required

Install Curlftps

Curlftps allows “mounting” and FTP site as a virtual filesystem, allowing navigating the FTP site the same as if it were an attached drive.

- Install:
`Pacman -S curlftps`
- Mount CV FTP (Network Solutions host):

```
curlftps ftp://004b4e7.netsolhost.com /mnt/ftp/ -o user=tomschlitz:Password321
```

Install VSFTP FTP Service

- To install:
`pacman -S vsftpd`
- Configure:

Enabling uploading

The `WRITE_ENABLE` flag must be set to YES in `/etc/vsftpd.conf` in order to allow changes to the filesystem, such as uploading:

```
write_enable=YES
```

Local user login

One must set the line to `/etc/vsftpd.conf` to allow users in `/etc/passwd` to login:

```
local_enable=YES
```

```
Also, set the line "anonymous_enable" = NO (default is YES)
```

- Start the service:

```
systemctl start vsftpd
```

- Start VSFTP on boot:
systemctl enable vsftpd

Enabling A Service At Startup

BE CAREFUL WITH THIS! Misconfiguration may cause loss of communication, forcing you to revert to original PogoPlug FW to access this file again (or putting the drive on another Linux machine). To configure a service to be automatically started at boot time, use the `systemctl` command in the following form:

```
systemctl enable service_name.service
```

The service will be started the next time you boot the system.

Set Static IP Address:

Set `/etc/network.d/wired-eth0` with (assumes router is at 192.168.1.1, desired static address is 222, and wireless does not extend to 222):

```
CONNECTION='ethernet'  
DESCRIPTION='A basic static ethernet connection using iproute'  
INTERFACE='eth0'  
IP='static'  
ADDR='192.168.1.222'  
#ROUTES=('192.168.0.0/24 via 192.168.1.2')  
GATEWAY='192.168.1.1'  
DNS=('192.168.1.1')  
  
## For IPv6 autoconfiguration  
#IP6=stateless  
  
## For IPv6 static address configuration  
#IP6='static'  
#ADDR6=('1234:5678:9abc:def::1/64' '1234:3456::123/96')  
#ROUTES6=('abcd::1234')  
#GATEWAY6='1234:0:123::abcd'
```

Could probably remove commented out lines, change the description, and otherwise clean up.

To install samba

Pacman -S samba

To start the service:

```
systemctl start smbd nmbd
```

or systemctl restart smbd nmbd to restart it after a change

To enable on start-up:

```
systemctl enable smbd nmbd
```

After install, copy /etc/samba/smb.conf.default to smb.conf, then edit this file.

Necessary changes:

- under [global], set workgroup = WORKGROUP
- set server string to something familiar (PogoPlug Server)
- under [public], uncomment, and set path = /mnt/backup (or name of dir or drive mount), for full public access.
- Other changes also – see latest on Pogoplug.

To install Subversion (SVN)

Pacman -S subversion

mkdir /var/lib/svnroot (assuming “/var/lib/svnroot” is the path to the repository root directory)

Either copy an existing repository, or create one using

```
svnadmin create /var/lib/svnroot/<repo name>
```

Edit the conf/svnserve.conf in the repository (if newly created) to set up password, etc.

Edit the file /etc/conf.d/svnserve to include the line SVNSERVE_ARGS="/var/lib/svnroot"

Start the server:

```
/usr/bin/svnserve -d -r /var/lib/svnroot
```

To stop the svn server, first find the process ID (2nd column) by:

```
ps auxww | fgrep svnserve
```

Then kill it using:

```
Kill <pid>
```

Configuration for a project, including the passwords, is in the /var/lib/svnroot/<project>/conf/svnserve.conf file. For the “projects” project, we used the passwd-team file, in the svnroot folder, by adding “password-db = /var/lib/svnroot/passwd-team” to the svnserve.conf file.

Mount External Drive

Mount it manually first:

`df -h` to see what drives are attached.

Or a better one: `lsblk`.

`mount /dev/sdc1 /mnt/WD_Elements_2TB` (device depends, name depends...)

Make it mount on boot:

Edit `/etc/fstab` (`vi /etc/fstab`) – *careful with this one, it can keep it from booting!*

```
# <file system> <dir> <type> <options> <dump> <pass>
/dev/sdc1 /mnt/WD_Elements_2TB ext2 defaults,noatime 0 0
```

NOTE: for some reason the PogoPlug doesn't boot properly with the external drive attached to the rear/lower USB connection. Unplug before rebooting.

To mount an NTFS drive (Windows), load this package

`Pacman -S ntfs-3g`

The mount command should then automatically recognize the filetype

YaBB Requirements

System Requirements to Install YaBB

- A web hosting account with a website
- Access to upload files to your website (standard on most hosts)
- Perl 5.8 (or higher) installed on the server your web account is on (standard on most hosts)
- CGI.pm v2.4+ and Socket.pm modules installed on the server your web account is on (standard on most hosts with Perl)
- At least 2 MB of disk space on your webspace account; over 10 MB recommended
- Sendmail (for Linux/Unix) or SMTP (for Windows/NT or Linux/Unix); optional, but recommended for sending emails

Install perl-net-smtp-ssl

This package is need for sending mail from Perl, using (through service???) Using xmail seems to work)

- First install its dependency: perl-io-socket-ssl

```
pacman -S perl-io-socket-ssl
```

- Install

```
Pacman -S
```

Virtual Hosts

This will describe how to serve multiple web pages from a single IP address, using ArchLinux.

1. Create a directory for each website to be hosted, as:

`/srv/http` for the main site and

`/srv/http_CMA` for the “CaptainMalApps” site.

2. Change `httpd_conf` (in `/etc/httpd/conf`): uncomment the line:

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

3. Change `httpd-vhosts.conf` to add a new `<VirtualHost *:80>` block for each virtual host (see example below).

4. Check syntax using “`apachectl configtest`”.

5. Change `hosts` file (in `/etc`). Append each entry with the host name, found in the `hostname` file (in `/etc`), as:

```
127.0.0.1 localhost.localdomain      localhost gordonserver1
::1          localhost.localdomain    localhost gordonserver1
```

6. Restart Apache using “`systemctl restart httpd`”.

Link to source article: <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-arch-linux>

Note that this article refers to an “`rc.d`” command, which we don’t have. It also shows quotes around the `DocumentRoot` entry, which makes it fail. So perhaps this isn’t such a good reference after all! Try:

<http://httpd.apache.org/docs/2.2/vhosts/examples.html>

To test, use “`apachectl configtest`”.

To restart Apache and make it all work, use “`systemctl restart httpd`”.

Example httpd-vhosts.conf file

This file is located in `/etc/httpd/conf/extra`

```
<VirtualHost *:80>
  ServerAdmin support@CaptainMalApps.com
  DocumentRoot /srv/http_CMA
  ServerName CaptainMalApps.com
  ServerAlias www.CaptainMalApps.com
  ErrorLog "/var/log/httpd/dummy-host2.example.com-error_log"
  CustomLog "/var/log/httpd/dummy-host2.example.com-access_log" common
</VirtualHost>
```

```
<VirtualHost *:80>
  ServerAdmin tom@barefootsoftwareconsulting.com
  DocumentRoot /srv/http
  ServerName BarefootSoftwareConsulting.com
  ServerAlias www.BarefootSoftwareConsulting.com
  ErrorLog "/var/log/httpd/dummy-host.example.com-error_log"
  CustomLog "/var/log/httpd/dummy-host.example.com-access_log" common
</VirtualHost>
```

```
<VirtualHost *:80>
  ServerAdmin tom@GordonInnovations.com
  DocumentRoot /srv/http
  ServerName GordonInnovations.com
  ServerAlias www.GordonInnovations.com
  ErrorLog "/var/log/httpd/dummy-host2.example.com-error_log"
  CustomLog "/var/log/httpd/dummy-host2.example.com-access_log" common
</VirtualHost>
```

Discussion Board

To Copy YaBB

- Copy the YaBB directory under the html/http directory
- Copy the YaBB directory under the cgi-bin directory
- Change the URL and server paths in Paths.pl
- **chmod** all directories to 777, all regular files to 666 and all *.pl and *.pm files to 755.
- Change the path to Perl in these files:
 - YaBB.pl:#! /bin/perl --
 - AdminIndex.pl:#! /bin/perl --
 - Setup.pl:#! /bin/perl -- (probably shouldn't need this one)*(see "useful commands", below for how to do this with just a few commands)*

Postfix (for mail)

This assumes MySQL has been installed (see Install MySQL, above).

- Install postfix: `pacman -S postfix`.

- Appendix

Useful File Locations

- Configuration for wired network is in `/etc/network.d/wired-eth0`. Examples for others (wireless, for example) are in `/etc/network.d/examples`.
- Apache configuration in `/etc/httpd/conf/httpd.conf`. Make files here rather than `.htaccess` when possible – supposed to be faster (`.htaccess` affects individual directories).
- Devices for mounting are in `/dev/`. Typical mount point is `/mnt/`.
- `/var/log` for system log files.

Useful Commands

- `/sbin/reboot` – reboots the system.
- `pacman -Syu` – updates all installed packages.
- `passwd` – change the current user’s password (`passwd <user>` to change other).
- `mount /dev/<device> /mnt/<folder>` allows reading from a mounted device, via the specified `/mnt/` folder; be sure to create the folder first! Example: `mount /dev/sdb1 /mnt/old`.
- `umount /mnt/<folder>` “un-mounts” the specified folder, mounted above.
- `systemctl start|stop|restart|enable` – start’s, stops, re-starts or enables (starts it at boot) a service, such as `vsftpd`, for example.
- `pacman -S <package>` to install a package on ArchLinux.
- `pacman -Q` shows all installed packages (may want to pipe to file).
- `useradd -m <username> -s /bin/bash <username>`
- `chgrp <username> <file or dir>`
- `chown <username> <file or dir>`
- `find -type f -name "*.pl" -ls`
.. to find all files of type `*.pl`, recursively, from the current directory, and display permissions.
- To update all files to 666 (read/write all), then all `*.pl` and `*.pm` to 755 (read/execute all, owner write), do:
 - `find -type f -name "*.*" -print0 |xargs -0 chmod 666`
 - `find -type f -name "*.pl" -print0 |xargs -0 chmod 755`

- `find -type f -name "*.pm" -print0 |xargs -0 chmod 755`
- To update all directories to 777 (read/write/execute all), do:
 - `find -type d -print0 |xargs -0 chmod 777`
- For a “task manager” real-time display, try “top”.
- `systemctl list-unit-files` to list active services.
- To backup up the entire drive to a disk image on another device (mounted as “sd”):

```
dd if=/dev/sda conv=sync,noerror bs=64K | gzip -c > /mnt/sd/root_img.gz
```

- To restore:

```
gunzip -c /mnt/sd/root_img.gz | dd of=/dev/sdb conv=sync,noerror bs=64K
```

- To clone an entire drive (MBR and all) (assuming empty flash is in sdb!):

```
dd if=/dev/sda of=/dev/sdb bs=4096 conv=notrunc,noerror
```

- For wiping a drive (say, second USB device):

```
dd if=/dev/zero of=/dev/sdb bs=1M
```

- For testing e-mail:

```
mailx -s test [mail address]
[message]
.
```

- To list all drives, even if not mounted (must run as root):
`Fdisk -l`
- To copy files from another machine (the `-r` option it to copy folders recursively:
`scp -r root@192.168.1.200:/var/www/* .`
Note, spaces must be “double-escaped”, as “some\\ path\\ name”.
- To get a list of all block (drive) devices, mounted or not: “lsblk”.
- To get a list of mounted drives, with human-readable space total/available: “df -h”.