Python: Dead man's switch

Nate Drake shows how to turn your Pi into a handy dead man's switch to send emails or update websites for you in case the worst happens.

Our expert

Nate Drake is a freelance tech journalist specialising in cybersecurity. His dead man's switch asks his long-suffering girlfriend to remember to water his rubber plants if he disappears.

COMPATIBILITY ALL Pis

Quick tip

This tutorial will work on any computer running a Debian based distro of Linux, so feel free to experiment with other computers if you wish.

Did you have the misfortune to sit through *Superman IV: A Quest for Peace*? If so, you'll remember a scene where the Man of Steel is forced to stop a subway train that has shot off down a tunnel. After single-handedly halting the train and saving everyone on board, Superman takes the time to reassure the public that the subway system is one of the safest ways to travel. This is in fact true, because while the fictional Metropolis may not make use of them, real-life subway trains have a dead man's switch—a control that has to be continually pressed in order for the train to move.

With the invention of inexpensive computers, it's now extremely easy to set up a digital dead man's switch. Wikileaks for instance has released a number of encrypted files of classified information, the password for which will be released if anything happens to any of the founders.

Traditionally, a dead man's switch is time-based, like Google's Inactive Account Manager, which allows you to nominate someone else to access your emails, documents and photos if your account is inactive for a period of time.

Not all applications of the switch concept are a safeguard against being arrested or dying. The website TimeCave, for instance, enables you to schedule an email at any point in the future up until 31 December 2025. This naturally can be anything from the location of your family's silverware to a few heartfelt messages for the people you love.

Dead man coding

In this tutorial we'll show you how to turn your Raspberry Pi into a dead man's switch. The Pi is perfect for this given the minimal amount of power it uses and the coding required.

In order to proceed, you'll need a Pi, ideally with a clean install of Raspbian on it. The Pi Zero is ideal for this purpose as the Dead Man's Switch is very low on system resources. This tutorial also assumes that you know how to connect to your Pi over SSH and know a few basic Terminal commands.

It's important before starting to decide what you wish your Pi to do. Perhaps you have an elderly relative who wants to check in with you regularly? Alternatively, you might want the Pi to delete your personal files if you fail to reset the switch after a certain amount of time.

The Dead Man's Switch itself works by way of a shell script which is regularly invoked by the scheduling utility *cron* on the Pi. The script will check to see if the period of time you specify (by default this is 24 hours) has lapsed and, if so, carries out your commands.

The script itself couldn't be simpler:

```
#!/bin/bash

# Dead Man's Script

timelimit=86400 #Time period before tripping switch in seconds.
lastaccessed=$(stat -c %Y $BASH_SOURCE)
timenow=$(date +%s)
let timeleft=($timenow-$lastaccessed)

if [ $timeleft -gt $timelimit ]
then
#place commands here
#sh ./ftp.sh
#sh ./email.sh
#shred -u /home/pi/mysecretfile.txt
fi

END_SCRIPT
exit 0
```

The script naturally has to be placed into a file and made executable on the Pi. (*See the box Set up your dead man's switch.*) This guide also includes templates for two very basic shell scripts which can be called by the Dead Man Switch script itself. One enables a web page to be updated to notify that the Dead Man's Switch has been tripped. Another will send out an e-mail to any address with the message of your choice. If you decide to use the FTP script, you will need access to a domain which allows access via FTP. There are a number of providers that will allow free web hosting plans if you don't have a domain of your own. The template for the email shell script makes use of a Gmail account. It makes sense to set one up specifically for the project.

In either case, if you wish to make use of these scripts, simply remove the # to uncomment the lines in the main switch script above. If you don't wish to broadcast the fact the dead man's switch has been tripped, it's just as feasible to protect your files with a password or run a backup. There's also a 'reset switch' script which will prevent the switch from being tripped for the next X hours. This would need to be executed periodically, for instance, by your dear old Nan to make sure an email isn't sent out to prompt you to come and check on her. Where there's a will there's a way.

For the sake of simplicity we'll assume that all scripts are located in the Pi's **home** folder, but you can place them elsewhere if you wish. By now the basic premise should be clear, and we strongly encourage you to adapt it to your own needs. If the reset switch is not executed, the script will keep executing itself every X hours. This means, for instance, that the script for the web page will keep updating to say the dead

Sendemail Simplicity

For the purposes of this guide, a Gmail account will be used to send an email when the switch is tripped, but the *sendemail* program can be used with any SMTP server provided you have the details. If you're happy to use Gmail, head over to gmail.com and create an account. Next head to **https://www.google.com/settings/security/lesssecureapps** to switch on access for 'less secure' apps.

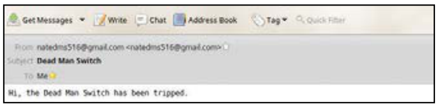
Next, on the Pi, you'll need to install the *sendemail* program and its dependencies

```
sudo apt-get install sendemail libio-socket-ssl-perl
```

and once this is all installed, type `nano email.sh` to create an empty file and paste in the following script, modifying it as needed:

```
#!/bin/bash
# Send e-mail scrippt
sendEmail -o tls=yes -f yourgmailaddress@gmail.com -t recipientsaddress@domain.com -s smtp.gmail.com:587 -xu yourgmailaddress@gmail.com -xp yourgmailpassword -u "Dead Man Switch" -m "Hi, This is an automated message from your Dead Man's Switch. The switch has been tripped."
END_SCRIPT
exit 0
```

Use Ctrl+x to exit, saving your changes.



From natedms516@gmail.com <natedms516@gmail.com>
Subject Dead Man Switch
To Me

Hi, the Dead Man Switch has been tripped.

› When setting up your email message, you may wish to come up with more original (and less Freudian) text to send yourself.

Next, run `chmod a+x email.sh` to make the FTP script executable.

Make sure to remove the `#` at the beginning of `#sh ./email.sh` in the main `script.sh` in order to invoke this if the switch is tripped.

man's switch has been tripped and list the date and time. This feature can be employed to create your own Warrant canary.

Warrant canaries have sprung up in the wake of 9/11 in response to shadowy government organisations placing secret subpoenas on companies to hand over information. In 2013, Ladar Levinson, the founder of the privacy-oriented email service Lavabit, came under pressure from the NSA to insert a secret backdoor into his own software, supposedly in a bid to track down Edward Snowden, who had previously used the service. Levinson was bound by law not to alert users that their data could be compromised by this, and so chose to shut down his service rather than undermine users' privacy.

Canny canary

As an alternative to shutting down or facing arrest for breaking the law, some companies choose to publish regular reports saying that as of this date, they have *not* been subject to such an order. If the company fails to confirm that they have not been served a secret warrant, this may not violate a gag order but users will know that their data isn't safe.

If you do decide to use the Pi as a Warrant Canary, you may wish to change the text of your web page or email to say words to the effect of 'Nothing to see here' or something more specific such as 'As of today we have received no warrants for one or more users' data.'

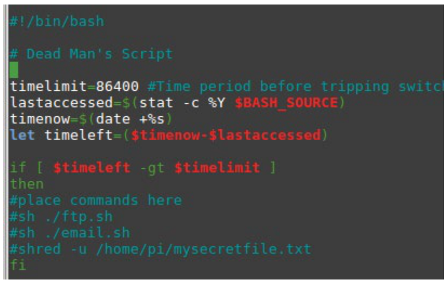
The Dead man's switch isn't foolproof. The legality of warrant canaries has yet to be tested in US/UK courts, and some countries such as Australia have banned them outright. The switch also needs to be physically secure, because if a burglar were to break into your old Granny's house, they could trip the 'reset' switch as easily as she could, ensuring that her SOS e-mail wouldn't be sent out. Also, if you're relying on the device to send an email or update a web page but it's disconnected from the internet for any reason, naturally the required action won't be carried out even though the switch is tripped. Additionally, if you're using your switch to delete data from a drive, it may be possible to use digital forensics to piece the data back together, so it's probably best to couple it with an email asking one of your loved ones to physically destroy your SD card/USB stick too.

Note: The FTP script doesn't encrypt data between the Pi and the website, so in theory someone could snoop on your connection and modify the website to suit themselves. You can circumvent this by using the sftp command or, if your domain provider allows, use SSH to connect to your website. LXF

Quick tip

When it comes to creating a warrant canary try to be imaginative. Instead of just posting some text or an image, why not regularly record a video to say that as of this date, you've received no warrants?

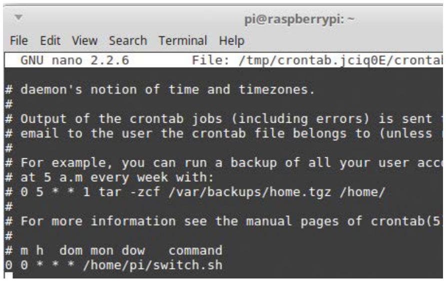
Set up your dead man's switch



```
#!/bin/bash

# Dead Man's Script

timelimit=86400 #Time period before tripping switc
lastaccessed=$(stat -c %Y $BASH_SOURCE)
timenow=$(date +%s)
let timeleft=($timenow-$lastaccessed)

if [ $timeleft -gt $timelimit ]
then
#place commands here
#sh ./ftp.sh
#sh ./email.sh
#shred -u /home/pi/mysecretfile.txt
fi
```

1 Create a switch script

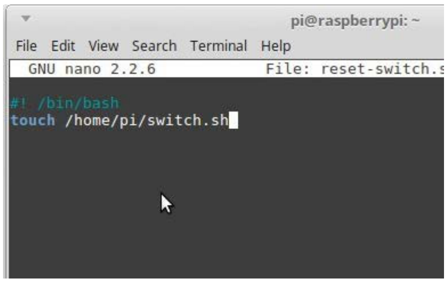
In your Pi's **home** folder, type `nano switch.sh` to create a file. Next paste in the script from the tutorial. (You may want to comment out lines to invoke FTP/e-mail scripts.) When you're done use Ctrl+x to quit, save, then run `chmod a+x script.sh` to execute the switch.



```
  GNU nano 2.2.6          File: /tmp/crontab.jciq0E/contal

# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent
# email to the user the crontab file belongs to (unless
#
# For example, you can run a backup of all your user acc
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5
#
# m h  dom mon dow   command
0 0 * * * /home/pi/switch.sh
```

2 Schedule a cron job

Simply run `crontab -e` to open the *Cron* scheduling utility which will act as our countdown timer. Choose Option 2 to edit using *nano* and at the very bottom of the file paste `0 0 * * * /home/pi/switch.sh/` . Use Ctrl+X to exit, saving your changes.



```
  GNU nano 2.2.6                File: reset-switch.s

#! /bin/bash
touch /home/pi/switch.sh
```

3 Create a reset switch

Type `nano reset-switch.sh` and paste in:

```
#! /bin/bash
touch /home/pi/switch.sh
```

Exit and type `chmod a+x reset-switch.sh` to make the reset switch executable. In future run `./reset-switch.sh` to reset the switch.