The paranoid ! Security Guide

sorcererś_apprentice

May 14, 2015

Abstract

Original thread is here.

Contents

1 Introduction

Hi all!

This is my first attempt to contribute something to the community. Basically you can find everything I write here somewhere else on the web or in some book - but exactly that is the problem. You can literally spend weeks digging up all this stuff. And to save you some trouble I thought: ”Heck, let’s just put this into a little manual.”

You’re dealing with a somewhat paranoid security setup for debian-based systems like **#!**.

(This is the end-user and *not* the —-—4xx0—2-version. We are not getting into virtual-virtual-virtual-machine-double-vpn-ssh-proxy-chain-from-your-internet-cafe-type-stuff.)

In this small guide I simply provide several ”recipes” for securing both your box and your internet-connection and web-applications. I won’t go into the why of all of this in too much detail as I want to provide a simple how-to that people can follow to make their system more secure without having to read through hundreds of pages of explanations. This information can easily be found elsewhere. If you’re interested in a certain topic then just fire up a web-search and give it a read.

This guide is not exhaustive of course. As they say, security is a process - and so this guide can only be a place to start which needs to be adjusted to your personal needs.

If you consider to use this information and you find something to be too overcautious for your particular need - just ignore it and move on. One last thing before we begin: I am not a ”security-guru” (far from

it) - but more appropriately (as my nick suggests) some dude wrapping his head around things...

2 BIOS-Passwords

For the physical security of your data you should always employ encrypted drives. But before we get to that make sure you set strong passwords in BIOS for *both* starting up and modifying the BIOS-settings. Also make sure to disable boot for any media other than your harddrive.

3 Encryption

With **#!** this is easy. In the installation you can simply choose to use an encrypted LVM. (For those of you who missed that part on installation and would still like to use an encrypted partition withouth having to reinstall: use these instructions to get the job done.) For other data, e.g. data you store on transportable media you can use TrueCrypt - which is better than e.g. dmcrypt for portable media since it is portable, too. You can put a folder with TrueCrypt for every OS out there on to the *un*encrypted part of your drive and thus make sure you can access the files everywhere you go. ‘

This is how it is done:

3.1 Making TrueCrypt Portable

- Download yourself some TC copy.
- Extract the tar.gz
- Execute the setup-file
- When prompted choose ”Extract .tar Package File”
- go to /tmp
- copy the tar.gz and move it where you want to extract/store it
- extract it
- once it’s unpacked go to ”usr”-¿”bin” grab ”truecrypt”-binary
- copy it onto your stick
- give it a test-run

There is really not much more in that tarball than the binary. Just execute it and you’re ready for some crypto.

I don't recommend using TrueCrypt's hidden container, though. Watch this vid to find out why. If you don't yet know how to use TrueCrypt check out this guide. (TrueCrypt's standard encryption is AES-256. This encryption is really good but there are ways to attack it and you don't know how advanced certain people already got at this. So when prompted during the creation of a TrueCrypt container use: *AES-Twofish-Serpent* and as hash-algorithm use *SHA-512*. If you're not using the drive for serious video-editing or such you won't notice a difference in performance. Only the encryption process when creating the drive takes a little longer. But we get an extra scoop of security for that...)

3.2 Hardware Encryption

There are three different types of hardware encrypted devices available, which are generally called: SED (Self Encrypting Devices)

- Flash-Drives (Kingston etc.)

- SSD-Drives (Samsung etc.)

- HD-Drives (WD, Hitachi, Toshiba etc.)

They all use AES encryption. The key is generated within the device's microprocessor and thus no crucial data - neither password nor key are written to the host system. AES is secure - and thus using these devices can give some extra protection.

But before you think that all you need to do is to get yourself one of these devices and you're safe - I have to warn you: You're not.

So let's get to the reasons behind that.

3.3 Attacks on Full-Disk-Encryption

Below we will have a look at a debian specific attack using a vulnerability common with encrypted LVMs.

But you need to be aware that all disk-encryption is generally vulnerable - be it software- or hardware-based. I won't go into details how each of them work exactly - but I will try to at least provide you with a short explanation.

For **software-based** disk-encryption there are these known attacks:

- **DMA-Attacks** (DMA/HDMI-Ports are used to connect to a running, locked machine to unlock it)

- **Cold-Boot-Attacks** (Keys are extracted from RAM after a cold reboot)

- **Freezing of RAM** (RAM is frozen and inserted into the attacker's machine to extratct the key)

- **Evil-Maid-Attacks** (Different methods to boot up a trojanized OS or some kind of software-keylogger)

For **hardware-based** disk-encryption there are similar attacks:

- **DMA-Attacks** (same as with SW-based encryption)

- **Replug-Attacks** (Drive's data cable is disconnected and connected to attacker's machine via SATA-hotplugging)

- **Reboot-Attacks** (Drive's data cable is disconnected and connected to attacker's machine after enforced reboot. Then the bios-password is circumvented through the repeated pressing of the F2- and enter-key. After the bios integrated SED-password has been disabled the data-cable is plugged into the attacker's machine. This only works on some machines.)

- **Networked-Evil-Maid-Attacks** (Attacker *steals* the actual SED and replaces it with another containing a tojanized OS. On bootup victim enters it's password which is subsequently send to the attacker via network/local attacker hot-spot. Different method: Replacing a laptop with a similar model (at e.g. airport/hotel etc.) and the attacker's phone# printed on the bottom of the machine. Victim boots up enters "wrong" password which is send to the attacker via network. Victim discovers that his laptop has been misplaced, calls attacker who now copies the content and gives the "misplaced" laptop back to the owner.)

A full explanation of all these attacks been be found in this presentation. (Unfortunately it has not yet been translated into English.) An English explanation of an evil-maid-attack against TrueCrypt encrypted drives can be found here

3.4 Attacks on encrypted Containers

There are also attacks against encrypted containers. They pretty much work like cold-boot-attacks, without the booting part.

An attacker can dump the container's password if the computer is either running or is in hibernation mode - either having the container open and even when the container has been opened during that session - using temporary and hibernation files.

3.5 Debian's encrypted LVM pwned

This type of "full" disk encryption can also be fooled by an attack that could be classified as a custom and extended evil-maid-attack. Don't believe me? Read this!

The problem basically is that although most of the filesystem and your personal data are *indeed* encrypted - your boot partition and GRUB aren't. And this allows an attacker with physical access to your

box to bring you into real trouble.

To avoid this do the following:MICAH LEE WROTE:

> If you dont want to reinstall your operating system, you can format your USB stick, copy /boot/* to it, and install grub to it. In order to install grub to it, youll need to unmount /boot, remount it as your USB device, modify /etc/fstab, comment out the line that mounts /boot, and then run grub-install /dev/sdb (or wherever your USB stick is). You should then be able to boot from your USB stick.
>
> An important thing to remember when doing this is that a lot of Ubuntu updates rewrite your initrd.img, most commonly kernel upgrades. Make sure your USB stick is plugged in and mounted as /boot when doing these updates. Its also a good idea to make regular backups of the files on this USB stick, and burn them to CDs or keep them on the internet. If you ever lose or break your USB stick, youll need these backups to boot your computer.
>
> One computer I tried setting this defense up on couldnt boot from USB devices. I solved this pretty simply by making a grub boot CD that chainloaded to my USB device. If you google Making a GRUB bootable CD-ROM, youll find instructions on how to do that. Heres what the menu.1st file on that CD looks like:
>
> default 0
>
> timeout 2
>
> title Boot from USB (hd1)
>
> root (hd1)
>
> chainloader +1
>
> I can now boot to this CD with my USB stick in, and the CD will then boot from the USB stick, which will then boot the closely watched initrd.img to load Ubuntu. A little annoying maybe, but it works.

(Big thanks to Micah Lee!)

Note: Apparently there is an issue with installing GRUB onto USB with waldorf/wheezy. As soon as I know how to get that fixed I will update this section.

3.6 Solutions

You might think that mixing soft- and hardware-based encryption will solve these issues. Well, no. They don't. An attacker can simply chain different methods and so we are back at square one. Of course this makes it harder for an attacker to reach his goals - but he/she will not be stopped by it. So the only method that basically remains is to regard full-disk-encryption as a first layer of protection only.

Please don't assume that the scenarios described above are somewhat unrealistic. In the US there are about 5000 laptops being lost or stolen each week on airports *alone*. European statistics indicate that about 8% of all business-laptops are at least once either lost or stolen.

A similar risk is there if you leave the room/apartment with your machine locked - but running. So the first protection against these methods is to always power down the machine. Always.

The next thing to remind yourself off is: You cannot rely on full-disk-encryption. So you need to employ further layers of encryption. That means that you will have to encrypt folders containing sensitive files again using other methods such as tomb or TrueCrypt. That way - if an attacker manages to get hold of your password he/she will only have access to rather unimportant files. If you have sensitive or confidential data to protect full-disk encryption is not enough!

When using encrypted containers that contain sensitive data you should shutdown your computer after having used them to clear all temporary data stored on your machine that could be used by an attacker to extract passwords.

If you have to rely on data being encrypted and would be in danger if anyone would find the data you were encrypting you should consider only using a power-supply when using a laptop - as opposed to running on power and battery. That way if let's say, you live in a dictatorship or the mafia is out to get you - and *they* are coming to your home or wherever you are - all you need to do when you sense that something weird is going on is to pull the cable and hope that *they* still need at least 30 secs to get to your ram. This can help prevent the above mentioned attacks and thus keep your data safely hidden.

3.7 eCryptfs

If for some reason (like performance or not wanting to type in thousands of passwords on boot) you don't want to use an encrypted LVM you can use ecryptfs to encrypt files and folders after installation of the OS.

To find out about all the different features of ecryptfs and how to use them I would like to point you to bodhi.zazen's excellent ecryptfs-tutorial.

But there is one thing that is also important for later steps in this guide and is generally a good idea to do:

3.8 Encrypting swap using ecryptfs

Especially when using older machines with less ram than modern computers it can happen quite frequently that your machine will use swap for different tasks when there's not enough ram available to

do the job. Apart from the lack of speed this is isn't very nice from a security standpoint: as the swap-partition is not located within your ram but on your harddrive - writing into this partion will leave traces of your activities on the harddrive itself. If your computer happens to use swap during your use of encryption tools it can happen that the passwords to the keys are written to swap and are thus extractable from there - which is something you really want to avoid.

You can do this very easily with the help of ecryptfs.

First you need to install it:

```
$ sudo apt-get install ecryptfs-utils cryptsetup
```

Then we need to actually encrypt our swap using the following command:

```
$ sudo ecryptfs-setup-swap
```

Your swap-partition will be unmounted, encrypted and mounted again.

To make sure that it worked run this command:

```
$ sudo blkid | grep swap
```

The output lists your swap partion and should contain "cryptswap".

To avoid error messages on boot you will need to edit your /etc/fstab to fit your new setup:

```
$ sudo geany /etc/fstab
```

Copy the content of that file into another file and save it. You will want to use it as back-up in case something gets screwed up.

Now make sure to find the entry of the above listed encrypted swap partition. If you found it go ahead and delete the other swap-entry relating to the unencrypted swap-partition. Save and reboot to check that everything is working as it should be.

3.9 Tomb

Another great crypto-tool is Tomb provided by the dyne-crew.

Tomb uses LUKS AES/SHA-256 and can thus be consider secure. But Tomb isn't just a possible *replacement* for tools like TrueCrypt.

It has some *really* neat and easy to use features:

1) Separation of encrypted file and key

2) Mounting files and folders in predefined places using bind-hooks

3) Hiding keys in picture-files using steganography

The documentation on Tomb I *was able to find*, frankly, seems to be scattered all over the place.

After I played around with it a bit I also came up with some tricks that I did not see being mentioned in any documentation.

And because I like to have everything in one place I wrote a short manual myself:

Installation:

First you will need to import dyne's keys and add them to your gpg-keylist:

```
$ sudo gpg -$\,$-fetch-keys \url{http://apt.dyne.org/software.pub}
```

Now verify the key-fingerprint.

```
$ sudo gpg -$\,$-fingerprint software@dyne.org | grep fingerprint
```

The output of the above command should be:

```
Key fingerprint = 8E1A A01C F209 587D 5706 3A36 E314 AFFA 8A7C 92F1
```

Now, after checking that you have the right key you can trust add it to apt:

```
$ sudo gpg -$\,$-armor -$\,$-export software@dyne.org > dyne.gpg
$ sudo apt-key add dyne.gpg
```

After you did this you want to add dyne's repos to your sources.list:

```
$ sudo geany /etc/apt/sources.list
```

Add:

```
deb \url{http://apt.dyne.org/debian} dyne main deb-src \url{http://apt.dyne.org
    /debian} dyne main
```

To sync apt:

```
$ sudo apt-get update
```

To install Tomb:

```
$ sudo apt-get install tomb
```

Usage:

If you have your swap activated Tomb will urge you to turn it off or encrypt it. If you encrypt it and leave it on you will need to include - -ignore-swap into your tomb-commands. To turn off swap for this session you can run

```
$ swapoff -a
```

To disable it completely you can comment out the swap in /etc/fstab. So it won't be mounted on reboot. (Please be aware that disabling swap on older computers with not much ram isn't such a good idea. Once your ram is being used fully while having no swap-partition mounted processes and programs will crash.)

Tomb will create the crypto-file in the folder you are currently in - so if you want to create a tomb-file in your documents-folder make sure to

```
$ cd /home/user/documents
```

Once you are in the right folder you can create a tomb-file with this command:

```
$ tomb -s XX create FILE
```

XX is used to denote the size of the file in MB. So in order to create a file named "test" with the size of 10MB you would type this:

```
$ tomb -s 10 create test
```

Please note that if you haven't turned off your swap you will need to modify this command as follows:

```
$ tomb -$\,$-ignore-swap -s 10 create test
```

To unlock and mount that file on /media/test type:

```
$ tomb open test.tomb
```

To unlock and mount to a different location:

```
$ tomb open test.tomb /different/location
```

To close *that particular* file and lock it:

```
$ tomb close /media/test.tomb
```

To close all tomb-files:

```
$ tomb close all
```

or simply:

```
$ tomb slam
```

After these basic operations we come to the fun part:

3.10 Advanced Tomb-Sorcery

Obviously having a file lying around somewhere entitled: ”secret.tomb” isn’t such a good idea, really.

A better idea is to make it harder for an attacker to even find the encrypted files you are using. To do this we will simply move its content to another file.

Example:

```
$ touch true-story.txt true-story.txt.key
$ mv secret.tomb true-story.txt
$ mv secret.tomb.key true-story.txt.key
```

Now you have changed the filename of the encrypted file in such a way that it can’t easily be detected.

When doing this you have to make sure that the filename syntax tomb uses is conserved:

```
filename.suffix filename.suffix.key
```

Otherwise you will have trouble opening the file.

After having hidden your file you might also want to move the key to another medium.

```
$ mv true-story.txt.key /medium/of/your/choice
```

Now we have produced quite a bit of obfuscation. Now let’s take this even further:

After we have renamed our tomb-file and separated key and file we now want to make sure our key can’t be found either.

To do this we will hide it within a jpeg-file.

```
$ tomb bury true-story.txt.key invisible-bike.jpg
```

You will need to enter a steganography-password in the process.

Now rename the original keyfile to something like ”true-story.txt.key-backup” and check if everything worked:

```
$ tomb exhume true-story.txt.key invisible-bike.jpg
```

Your key should have reappeared now. After making sure that everything works you can safely bury the key again and delete the residual key that usually stays in the key’s original folder.

By default Tomb’s encrypted file and key need to be in one folder. If you have separated the two you will have to modify your opening-command:

```
$ tomb -k /medium/of/your/choice/true-story.txt.key open true-story.txt
```

To change the key-files password:

```
$ tomb passwd true-story.txt.key
```

If, let's say, you want to use Tomb to encrypt your icedove mail-folders you can easily do that. Usually it would be a pain in the butt to do this kind of stuff with e.g. truecrypt because you would need to setup a container, move the folder to the container and when using the folder you would have to move back to its original place again.

Tomb does this with ease:

Simply move the folders you want to encrypt into the root of the tomb-file you created.

Example:

You want to encrypt your entire .icedove folder. Then you make a tomb-file for it and move the .icedove folder into that tomb. The next thing you do is create a file named "bind-hooks" and place it in the same dir. This file will contain a simple table like this:

```
.icedove .icedove .folder-x .folder-x .folder-y .folder-y .folder-z .folder-z
```

The fist column denotes the path relative to the tomb's root. The second column represents the path relative to the user's home folder.

So if you simply wanted to encrypt your .icedove folder - which resides in /home/user/ the above notation is fine. If you want the folder to be mounted elsewhere in the your /home you need to adjust the lines accordingly.

One thing you need to do after you moved the original folder into the tomb is to create a dummy-folder into which the original's folders content can be mounted. So you simply go into /home/user and create a folder named ".icedove" and leave it empty.

The next time you open and mount that tomb-file your .icedove folder will be where it should be and will disappear as soon as you close the tomb. Pretty nice, hu?

I advise to test this out before you actually move all your mails and prefs into the tomb. Or simply make a backup. But use some kind of safety-net in order not to screw up your settings.

4 Keyloggers

Keyloggers can pose a great thread to your general security - but especially the security of your encrypted drives and containers. If someone manages to get a keylogger onto your system he/she will

be able to collect all the keystrokes you make on your machine. Some of them even make screen-shots.

So what kind of keyloggers are there?

4.1 Software Keyloggers

For linux there are several software-keyloggers available. Examples are lkl, uberkey, THC-vlogger, PyKeylogger, logkeys.

4.2 Defense against Software Keyloggers

1) **Never use your system-passwords outside of your system**

Generally everything that is to be installed under linux needs root access or some priveliges provided through /etc/sudoers. But an attacker could have obtained your password if he/she was using a browser-exploitation framework such as beef - which also can be used as a keylogger on the browser level. So if you have been using your sudo or root password anywhere on the internet it might have leaked and could thus be used to install all kinds of evil sh*t on your machine. Keyloggers are also often part of rootkits. So do regular system-checks and use intrusion-detection-systems.

2) **Make sure your browser is safe**

Often people think of keyloggers only as either a software tool or a piece of hardware equipment installed on their machine. But there is another threat that is actually much more dangerous for linux users: a compromised browser. You will find a lot of info on how to secure your browser further down. So make sure you use it.

Compromising browsers isn't rocket science. And since all the stuff that is actually dangerous in the browser is cross-plattform - you as a linux-user aren't safe from that. No matter what short-sighted linux-enthusiasts might tell you. A java-script exploit *will* pwn you - if you don't secure your browser. No matter if you are on OSX, Win or debian.

3) **Check running processes**

If your attacker isn't really skilled or determined he/she might not think about hiding the process of the running keylogger. You can take a look at the output of

```
$ ps -aux or
$ htop or
$ pstree
```

and inspect the running processes. Of course the attacker could have renamed it. So have a look for suspicious processes you have never heard of before. If in doubt do a search on the process or ask in a security-related forum about it.

Since a lot of keyloggers come as the functionality of a rootkit it would be much more likely that you would have one of these.

4) **Do daily scans for rootkits**

I will describe tools for doing that further below. RKHunter and chkrootkit should definitely be used. The other IDS-tools described give better results and are much more detailed - but you actually need to know a little about linux-architecture and processes to get a lot out of them. So they're optional.

5) **Don't rely on virtual keyboards**

The idea to defeat a keylogger by using a virtual keyboard is nice. But is also dangerous. There are some keyloggers out there that will also capture your screen activity. So using a virtual keyboard is pretty useless and will only result in the false feeling of security.

4.3 Hardware Keyloggers

There is also an ever growing number of hardware keyloggers. Some of which use wifi. And some of them can be planted inside your keyboard so you wouldn't even notice them if you inspected your hardware from the outside.

4.4 Defense against Hardware Keyloggers

1) **Inspect your Hardware**

This one's obvious.

2) **Check which devices are connected to your machine**

There is a neat little tool called USBView which you can use to check what kind of usb-devices are connected to your machine. Some - *but not all* - keyloggers that employ usb will be listed there. It is available through the debian-repos.

```
$ sudo apt-get install usbview
```

Apart from that there's not much you can do about them. If a physical attack is part of your thread-model you might want to think about getting a laptop safe in which you put the machine when not in use or if you're not around. Also, don't leave your laptop unattended at work, in airports, hotels and on conferences.

5 Secure File-Deletion

Additional to encrypted drives you may also want to securely delete old data or certain files. For those who do not know it: regular ”file deletion” does not erase the ”deleted” data. It only unlinks the file’s inodes thus making it possible to recover that ”deleted” data with forensic software.

There are several ways to securely delete files - depending on the filesystem you use. The easiest is:

5.1 BleachBit

With this little tool you can not only erase free disc space - but also clean your system from various temporary files you don’t need any longer and that would give an intruder unnecessary information about your activities.

To install:

```
$ sudo apt-get install bleachbit
```

to run:

```
$ bleachbit
```

Just select what you need shredding. Remember that certain functions are experimental and may cause problems on your system. But no need to worry: BleachBit is so kind to inform you about that and give you the chance to cancel your selection.

Another great (and much more secure) tool for file deletion is:

5.2 srm (secure remove)

```
$ sudo apt-get install secure-delete
```

Usage:

```
Syntax: srm (-dflrvz) file1 file2 etc. Options: -d ignore the two dot special
    files "." and "..". -f fast (and insecure mode): no /dev/urandom, no
    synchronize mode. -l lessens the security (use twice for total insecure mode
    ). -r recursive mode, deletes all subdirectories. -v is verbose mode. -z
    last wipe writes zeros instead of random data.
```

5.3 Other ways to securely wipe drives

To overrite data with zeros:

```
# dd if=/dev/zero of=/dev/sdX or:
$ sudo dd if=/dev/zero of=/dev/sdX
```

To overwrite data with random data (makes it less obvious that data has been erased):

```
# dd if=/dev/urandom of=/dev/sdX or:
$ sudo dd if=/dev/urandom of=/dev/sdX
```

Note: shred doesn't work reliably with ext3.

6 Your Internet-Connection

Generally it is advised to use a *wired* LAN-connection - as opposed to wireless LAN (WLAN).

For further useful information in regards to wireless security read this. If you *must* use WLAN please use WPA2 encryption. Everything else can be h4xx0red by a 12-year-old using android-apps such as anti.

Another thing is: Try only to run services on your machine that you really use and have configured properly. If e.g. you don't use SSH - deinstall the respective client to make sure to save yourself some trouble. Please note that IRC also is not considered to be that secure. Use it with caution or simply use a virtual machine for stuff like that.

If you do use SSH please consider using Denyhosts or SSHGuard. (If you want to find out what might happen if you don't use such protection see foozer's post.)

So, let's begin with your firewall. For debian-like systems there are several possible firewall-setups and different guis to do the job. However, I found ipkungfu (an iptables-script) to do the best job while being easy to set up. This is how you set it up:

6.1 ipkungfu (basic configuration)

download and install:

```
$ sudo apt-get install ipkungfu
```

configure:

```
$ sudo geany /etc/ipkungfu/ipkungfu.conf
```

uncomment (and adjust):

```
# IP Range of your internal network. Use "127.0.0.1"
# for a standalone machine. Default is a reasonable
# guess. LOCAL_NET="192.168.1.0/255.255.255.0" -$\,$-$\,$-
# Set this to 0 for a standalone machine, or 1 for
# a gateway device to share an Internet connection.
# Default is 1. GATEWAY=0 -$\,$-$\,$-
# Temporarily block future connection attempts from an
# IP that hits these ports (If module is present) FORBIDDEN_PORTS="135 137 139"
    -$\,$-$\,$-
# Drop all ping packets?
# Set to 1 for yes, 0 for no. Default is no. BLOCK_PINGS=1 -$\,$-$\,$-
# What to do with 'probably malicious' packets
#SUSPECT="REJECT" SUSPECT="DROP" -$\,$-$\,$-
# What to do with obviously invalid traffic
# This is also the action for FORBIDDEN_PORTS
#KNOWN_BAD="REJECT" KNOWN_BAD="DROP" -$\,$-$\,$-
# What to do with port scans
#PORT_SCAN="REJECT" PORT_SCAN="DROP"
```

enable ipkungfu to start with the system:

```
$ sudo geany /etc/default/ipkungfu
```

change: "IPKFSTART = 0" ---¿ "IPKFSTART=1"

start ipkungfu:

```
$ sudo ipkungfu
```

fire up GRC's Shields Up! and check out the awesomeness.

(special thanks to the ubuntu-community)

6.2 Configuring /etc/sysctl.conf

Here you set different ways how to deal with ICMP-packets and other stuff:

```
$ sudo geany /etc/sysctl.conf
```

```
# Do not accept ICMP redirects (prevent MITM attacks) net.ipv4.conf.all.
   accept_redirects=0 net.ipv6.conf.all.accept_redirects=0 net.ipv4.
   tcp_syncookies=1
```

```
#lynis recommendations
#net.ipv6.conf.default.accept_redirects=0 net.ipv4.tcp_timestamps=0 net.ipv4.
    conf.default.log_martians=1
# TCP Hardening - (url)\url{http://www.cromwell-intl.com/security/security-
    stack-hardening.html(/url)} net.ipv4.icmp_echo_ignore_broadcasts=1 net.ipv4.
    conf.all.forwarding=0 net.ipv4.conf.all.rp_filter=1 net.ipv4.
    tcp_max_syn_backlog=1280 kernel.core_uses_pid=1 kernel.sysrq=0
#ignore all ping net.ipv4.icmp_echo_ignore_all=1
# Do not send ICMP redirects (we are not a router) net.ipv4.conf.all.
    send_redirects = 0
# Do not accept IP source route packets (we are not a router) net.ipv4.conf.all
    .accept_source_route = 0 net.ipv6.conf.all.accept_source_route = 0
# Log Martian Packets net.ipv4.conf.all.log_martians = 1
```

After editing do the following to make the changes permanent:

```
sudo sysctl -p
```

(thanks to tradetaxfree for these settings)

6.3 Modem & Router

Please don't forget to enable the firewall features of your modem (and router), disable UPnP and change the usernames and admin-passwords. Also try to keep up with the latest security info and updates on your firmware to prevent using equipment such as this. You might also want to consider setting up your own firewall using smoothwall.

Here you can run a short test to see if your router is vulnerable to UPnP-exploits.

The best thing to do is to use after-market-open-source-firmware for your router such as dd-wrt, openwrt or tomato. Using these you can turn your router into an enterprise grade device capable of some *real* Kungfu. Of course they come with heavy artillery - dd-wrt e.g. uses an IP-tables firewall which you can configure with custom scripts.

7 Intrusion-Detection, Rootkit-Protection & AntiVirus

7.1 snort (basic configuration)

The next thing you might want to do is to take a critical look at who's knocking at your doors.

For this we use snort. The setup is straight forward and simple:

```
$ sudo apt-get install snort
```

run it:

```
$ snort -D (to run as deamon)
```

to check out packages live type:

```
$ sudo snort
```

Snort should automatically start on reboot.

If you want to check out snort's rules take a look at: /etc/snort/rules

To take a look at snorts warnings:

```
$ sudo geany /var/log/snort/alert
```

Snort will historically list all the events it logged.

There you will find nice entries like this...

```
(**) (1:2329:6) MS-SQL probe response overflow attempt (**) (Classification:
    Attempted User Privilege Gain) (Priority: 1) (Xref => (url)\url{http://www.
    securityfocus.com/bid/9407)(/url)}
```

...and will thank the flying teapot that you happen to use **#!**

7.2 RKHunter

The next thing to do is to set up RKHunter - which is short for (R)oot(K)itHunter.

What does it do? You guessed it: It hunts down rootkits.

Installation again is simple:

```
$ sudo apt-get install rkhunter
```

The best is to run rkhunter on a clean installation - just to make sure nothing has been tampered with already.

One very important thing about rkhunter is that you need to give it some feedback: everytime you e.g. make an upgrade to your sytem and some of your binaries change rkhunter will weep and tell you you've been compromised. Why? Because it can only detect suspicious files and file-*changes*. So, if you go about and e.g. upgrade the coreutils package a lot of change will be happening in /usr/bin - and when you subsequently ask rkhunter to check your system's integrity your log file will be all red with warnings. It will tell you that the file-properties of your binaries changed and you start freaking out. To

avoid this simply run the command rkhunter - -propupd *on a system which you trust to not have been compromised*.

In short: directly after commands like apt-get update && apt-get upgrade run:

```
$ sudo rkhunter -$\,$-propupd
```

This tells rkhunter: 'sall good.

To run rkhunter:

```
$ sudo rkhunter -c -$\,$-sk
```

You find rkhunter's logfile in /var/log/rkhunter.log. So when you get a warning you can in detail check out what caused it.

To set up a cronjob for RKHunter:

```
$ sudo geany /etc/cron.daily/rkhunter.sh
```

insert and change the mail-address:

```
#!/bin/bash /usr/local/bin/rkhunter -c -$\,$-cronjob 2>&1 | mail -s "RKhunter
    Scan Details" your@email-address.com
```

make the script executable:

```
$ sudo chmod +x /etc/cron.daily/rkhunter.sh
```

update RKHunter:

```
$ sudo rkhunter -$\,$-update
```

and check if it functions the way it's supposed to do:

```
$ sudo rkhunter -c -$\,$-sk
```

Of course you can leave out the email-part of the cronjob if you don't want to make the impression on someone shoulder-surfing

your email-client that the only one who's sending you emails is your computer...

Generally, using snort and rkhunter is a good way to become paranoid - if you're not already. So please take the time to investigate the alerts and warnings you get. A lot of them are false positives and the listings of your system settings. Often enough nothing to worry about. But if you want to use them as security tools you will have to invest the time to learn to interpret their logs. Otherwise just skip them.

7.3 RKHunter-Jedi-Tricks

If you're in doubt whether you did a rkhunter - -propupd after an upgrade and you are getting a warning you can run the following command:

```
$ sudo rkhunter -$\,$-pkgmgr dpkg -c -$\,$-sk
```

Now rkhunter will check back with your package-manager to verify that all the binary-changes were caused by legitimate updates/upgrades. If you previously had a warning now you *should* get zero of them. If you still get a warning you can check which package the file that caused the warning belongs to.

To do this:

```
$ dpkg -S /folder/file/in/doubt
```

Example:

```
$ dpkg -S /bin/ls
```

Output:

```
coreutils: /bin/ls
```

This tells you that the file you were checking (in this case /bin/ls) belongs to the package "coreutils".

Now you can fire up packagesearch.

If you haven't installed it:

```
$ sudo apt-get install packagesearch
```

To run:

```
$ sudo packagesearch
```

In packagesearch you can now enter coreutils in the field "search for pattern". Then you select the package in the box below. Then you go over to the right and select "files". There you will get a list of files belonging to the selected package. What you want to do now is to look for something like:

```
/usr/share/doc/coreutils/changelog.Debian.gz
```

The idea is to get a file belonging to the same package as the file you got the rkhunter-warning for - but that is not located in the binary-folder.

Then you look for that file within the respective folder and check the file-properties. When it was modified at the same time as the binary in doubt was modified you can be quite certain that the change was

caused by a legitimate update. I think it is save to say that some script-kiddie trying to break into your system will not be that thorough. Also make sure to use debsums when in doubt. I will get to that a little further down.

Another neat tool with similar functionality is:

7.4 chkrootkit

To install:

```
$ sudo apt-get install chkrootkit
```

To run:

```
$ sudo chkrootkit
```

Other nice intrusion detection tools are:

7.5 tiger

Tiger is more thorough than rkhunter and chkrootkit and can aid big time in securing your box:

```
$ sudo apt-get install tiger
```

to run it:

```
$ sudo tiger
```

you find tiger's logs in /var/log/tiger/

7.6 Lynis

If you feel that all the above IDS-tools aren't enough - I got something for you:

LynisLYNIS WROTE:

> Lynis is an auditing tool for Unix (specialists). It scans the system and available software, to detect security issues. Beside security related information it will also scan for general system information, installed packages and configuration mistakes.
>
> This software aims in assisting automated auditing, software patch management, vulnerability and malware scanning of Unix based systems

I use it. It is great. If you think you might need it - give it a try. It's available through the debian repos.

```
$ sudo apt-get install lynis
```

To run:

```
$ sudo lynis -c
```

Lynis will explain its findings in the log-file.

7.7 debsums

debsums checks the md5-sums of your system-files against the hashes in the respective repos.

Installation:

```
$ sudo apt-get install debsums
```

To run:

```
$ sudo debsums -ac
```

This will list all the files to which the hashes are either missing or have been changed. But please don't freak out if you find something like: /etc/ipkungfu/ipkungfu.conf after you have been following this guide...

7.8 sha256

There are some programs that come with sha256 hashes nowadays. For example: I2P

debsums won't help with that. To check these hashes manually:

```
$ cd /folder/you/downloaded/file/to/check/to -sha256sum -c file-you-want-to-
    check
```

Then compare it to the given hash. Note: This tool is already integrated to debian-systems.

7.9 ClamAV

To make sure eveything that gets into your system is clean and safe use ClamA(nti)V(irus).

To install:

```
$ sudo apt-get install clamav
```

To update:

```
$ sudo freshclam
```

To inspect e.g. your download folder:

```
$ sudo clamscan -ri /home/your-username/downloads
```

This will ClamAV do a scan *recursively*, i.e. also scan the content of folders and inform you about possibly *infected files*.

To inspect your whole system:

```
$ sudo clamscan -irv -$\,$-exclude=/proc -$\,$-exclude=/sys -$\,$-exclude=/dev
    -$\,$-exclude=/media -$\,$-exclude=/mnt
```

This will make ClamAV scan your system recursively in verbose mode (i.e. show you what it is doing atm) whilst excluding folders that shouldn't be messed with or are not of interest and spit out the possibly infected files it finds. To also scan attached portable media you need to modify the command accordingly.

Make sure to test everything you download for possible infections. You never know if servers which are normally trustworthy haven't been compromised. Malicious code can be hidden in every usually employed filetype. (Yes, including .pdf!)

Remember: ClamAV is known for its tight nets. That means that you are likely to get some false positives from time to time. Do a web-search if you're in doubt in regards to its findings.

After you set up your host-based security measures we can now tweak our online security.

Starting with:

8 DNS-Servers

8.1 Using secure and censor-free DNS

To make changes to your DNS-settings:

```
$ sudo geany /etc/resolv.conf
```

change your nameservers to trustworthy DNS-Servers. Otherwise your modem will be used as "DNS-Server" which gets its info from your ISP's DNS.

And nah... We don't trust the ISP...

Here you can find secure and censor-free DNS-servers. The Germans look here.

HTTPS-DNS is generally preferred for obvious reasons.

Your resolv.conf should look something like this:

```
nameserver 213.73.91.35 #CCC DNS-Server nameserver 85.214.20.141 #FoeBud DNS-
    Server
```

Use *at least two* DNS-Servers to prevent connectivity problems when one server happens to be down or experiences other trouble.

To prevent this file to be overwritten on system restart fire up a terminal as root and run:

```
$ sudo chattr +i /etc/resolv.conf
```

This will make the file unchangeble - even for root.

To revoke this for future changes to the .conf run:

```
$ sudo chattr -i /etc/resolv.conf
```

This forces your web-browser to use the DNS-servers *you* provided instead of the crap your ISP uses.

To test the security of your DNS servers go here.

8.2 DNScrypt

What you can also do to secure your DNS-connections is to use DNScrypt.

The thing I don't like about DNScrypt is one of its core functions: to use OpenDNS as your resolver. OpenDNS has gotten quite a bad rep in the last years for various things like aggressive advertising and hijacking google-searches on different setups. I tested it out yesterday and couldn't replicate these issues. But I am certain that some of these "features" of OpenDNS have been actively blocked by my Firefox-setup (which you find below). In particular the addon Request Policy seems to prevent to send you to OpenDNS' search function when you typed in an address it couldn't resolve. The particular issue about that search function is that it apparently is powered by yahoo! and thus yahoo! would log the addresses you are searching for.

Depending on your threat-model, i.e. if you don't do anything uber-secret you don't want anybody to know, you might consider using DNScrypt, as the tool seems to do a good job at encrypting your DNS-traffic. There also seems to be a way to use DNScrypt to tunnel your queries to a DNS-server other than OpenDNS - but I haven't yet checked the functionality of this.

So, if you don't mind that OpenDNS will know **every** website you visit you might go ahead and configure DNScrypt:

Download the current version.

Then:

```
$ sudo bunzip2 -cd dnscrypt-proxy-*.tar.bz2 | tar xvf -
$ cd dnscrypt-proxy-*
```

Compile and install:

```
$ sudo ./configure && make -j2
$ sudo make install
```

Adjust -j2 with the number of cpu-cores you want to use for the compilation or have at your disposal.

Go and change your resolv.conf to use localhost:

```
$ geany /etc/resolv.conf
```

Modify to:

```
nameserver 127.0.0.1
```

Run DNScrypt as daemon:

```
$ sudo dnscrypt-proxy -$\,$-daemonize
```

According to the developer:JEDISCT1 WROTE:

> DNSCrypt will chroot() to this user's home directory and drop root privileges for this user's uid as soon as possible.

I have to admit that OpenDNS is really fast. What you could do is this: You could use OpenDNS for your ”normal” browsing. When you start browsing for stuff that you consider to be private for whatever reasons change your resolv.conf back to the trustworthy DNS-servers mentioned above - which you conveniently could keep as a backup file in the same folder. Yeah, that isn't slick, I know. If you come up with a better way to do this let me know. (As soon as I checked DNScrypt's function to use the same encryption for different DNS-Servers I will make an update.)

The next thing on our list is:

9 Firefox/Iceweasel

9.1 Firefox-Sandbox: Sandfox

Sandfox is a neat little script provided by IgnorantGuru which runs firefox (and other applications) in a sandboxed environment which prevents firefox's access to crucial filesystem-areas in case it gets compromised.

To install:

```
$ sudo -s
$ gpg -$\,$-keyserver keys.gnupg.net -$\,$-recv-keys 7977070
    A723C6CCB696C0B0227A5AC5A01937621
$ gpg -$\,$-check-sigs 0x01937621
$ bash -c 'gpg -$\,$-export -a 01937621 | apt-key add -'
$ echo "deb (url)\url{http://ignorantguru.github.com/debian/(/url)} unstable
    main" >> /etc/apt/sources.list
$ apt-get update
$ apt-get install sandfox
```

(Thanks to tradetaxfree)

To run:

```
$ sudo sandfox firefox
```

Type "/" into firefox address-bar to check out whether it works. Firefox should now only have access to files it really needs to function and not e.g. /root.

To be able to download stuff from the web you need to add a bind in sandfox's default profile:

```
$ sudo geany /etc/sandfox/default.profile
```

add:

```
bind=/home/$user/downloads
```

Check your systems filename-capitalization to make sure you really grant sandfox access to the right folder

In **#!** you can easily set this configuration as your default: simply go to "settings"-¿"openbox"-¿"GUI Menu Editor"-¿"Openbox"-¿"Web Browser". Then simply add the command "sandfox firefox". For this to work you need to once run

```
$ sudo sandfox firefox
```

after a system start to create a sandbox. If you happen to find this too much hassle simply go with tradetaxfree's init-script.

After you successfully sandboxed your browser we now continue to make that particular application *much* more secure than it is by default.

First go to:

9.2 Firefox-Preferences

and change these settings:

(Some of these are defaults already - but depending on who was/is using the machine you access the interwebs with and other varying factors you might want to control these settings.)

```
"General"->"when Firefox starts"->"Show a blank page"
"General"->"save files to:"Downloads"
"Content"->check:"Block pop-up windows"
"Content"->uncheck:"Enable JavaScript" (optional - NoScript Add-on will block
    it anyway)
"Content"->"Fonts & Colors"->"Advanced"->"Serif":"Liberation Sans"
"Content"->"Fonts & Colors"->"Advanced"->"Sans-serif":"Liberation Sans"
"Content"->"Fonts & Colors"->"Advanced"->uncheck:"Allow pages to choose their
    own fonts"
"Content"->"Languages"->choose *only*:"en-us" (remove all others, if any)
"Applications"->choose:"Always ask" for every application
    if not possible:choose:"Preview in Firefox/Nightly"
"Privacy"->"Tracking"->check:"Tell websites I do not want to be tracked"
"privacy"->"History"->"Firefox will:"Use custom settings for history"
"privacy"->"History"->uncheck:"Always use private browsing mode"
"privacy"->"History"->uncheck:"Remember my browsing and download history"
"privacy"->"History"->uncheck:"Remember search and form history"
"privacy"->"History"->uncheck:"Accept cookies from sites"
"privacy"->"History"->uncheck:"Accept third-party cookies"
"privacy"->"History"->check:"Clear history when Firefox/Nightly closes"
"privacy"->"History"->"settings":check all -> except:"Site Preferences"
    to enable cookies for certain trusted sites: use:"Exceptions" and paste URL
    of site and modify settings according to your preference.
    If you additionally use Cookie-Monster (Add-on) you need
    to uncheck "Block all cookies" in CM-Options
```

```
"privacy"->"location bar"->"When using the location bar, suggest:"->choose:"
    Nothing"
"security"->check:"Warn me when sites try to install add-ons"
"security"->check:"Block reported attack sites"
"security"->check:"Block reported web forgeries"
"security"->"Passwords"->uncheck:"Remember passwords for sites"
"security"->"Passwords"->uncheck:"Use a master password"
"advanced"->"General"->"System Defaults"->uncheck:"Submit crash reports"
"advanced"->"General"->"System Defaults"->uncheck:"Submit performance data"
"advanced"->"Update"->check:"Automatically install updates"
"advanced"->"Update"->check:"Warn me if this will disable any of my add-ons"
"advanced"->"Update"->check:"Automatically update Search Engines"
"advanced"->"Encryption"->"Protocols"->check:"Use SSL 3.0"
"advanced"->"Encryption"->"Protocols"->check:"Use TLS 1.0"
"advanced"->"Encryption"->"Certificates"->
    "When a server requests my personal certificate"->check:"Ask me every time"
```

9.3 Plugins

at the most use:

Java

Flash (Be aware of the latest security holes in flash!

Only allow them to run on *trusted* sites!

9.4 Addons

- Empty Cache Button (optional)
- Calomel SSL Validation (cool little addon which does exactly what its name says and also has some more tweaks in the settings)
- Adblock Edge (Filter Supscriptions: make sure you get some anti-tracking filters up and running! (depending on location & internet use))
- Easylist
- EasyPrivacy
- fanboy-adblock

- Fanboy's Tracking List
- Fanboy's Annoyance List
- BetterPrivacy (LSO/Flash-Cookie-Protection)
- Cookie Monster (Allows you to Manage your Cookie-Policies. For less baggage use Firefox/Iceweasel "Preferences" -¿ "Privacy")
- HTTPS-Everywhere (Download via EFF.org) (settings: enable SSL-Observatory but don't allow to transmit ISP-data)
- HTTPS Finder
- NoScript (go to "settings" and check "also apply on whitelisted sites")
- Perspectives (SSL-Cerfiticate-Control - go to settings: "notary servers" -¿ check "only contact when websites cause security error")
- RefControl (controls your HTTP-Referers - setting: "block" -¿ "3rd parties only")
- Request Policy (rejects cross-site requests)
- WOT (Web of Trust - user based website ratings that show up in websearches. Caution: Not very accurate. Always double check when in doubt. This addon tends

  to get abused by different groups of users who either give malicious sites good ratings - or flag perfectly good sites.)
- PwdHash (Nice addon to help your password management. Use "F2" when entering a password into a password field when setting up a new account somewhere to create a MD5-hash using your password and the domain. (When logging in you have to select the password-field and press F2 again to run the hashing.) This way you can use the same password on different sites without having to worry about security implications - because every site gets its own password generated through the hash. The tool is provided by Standford University and can be trusted. No data is actually transmitted to their servers. The hash is generated using your local java-script. If you need to login from a machine that doesn't have pwdhash installed: go to https://www.pwdhash.com/ -¿ their SSL is *very* strong.)
- FoxyProxy (a convenient Proxy Switcher)
- Useragent Switcher Does exactly that. But be careful: If you set your user-agent as shown below - using this addon it will overwrite these settings and will not automatically restore them if you turn off the switcher. So you would have to manually reconfigure about:config again. Which kinda sucks. But you can get a whole load really cool user agents here. Simply download the .xml and import it to the Useragent Switcher. There are really neat current agents in there: e.g. all kinds of

different web browser for all OSs and of course various bots. Google bot comes in handy when you need access to some forum...

- Web Developer (Has some cool features. If you like inspecting websites just check it out.)
- Bloody Vikings (Creates disposable mail-addresses)
- **Note:** You don't need Ghostery. The above mentioned Adblock lists do a much better job protecting you from web-tracking without using the additional resourced Ghostery uses .

Of course there are more addons you could use. But I don't really see the point of them. Most of them either are snake-oil or even dangerous. But please inform me if you happen to come across something *really* cool which could help improve security which none of the setting provided here can do. To keep your ISP and possible MITM-attackers from reading what you do on the web *always* use SSL - as far as it is available. To help with this use:

9.5 SSL-Search Engines

To get them go here.

The user SSL Search Bar has provided easily installable SSL-searchbar-plugins

You get SSL-plugins for all the alternative search-engines like ixquick, duckduckgo etc. there. Install those you happen to use.

Privatelee also looks promising. But I haven't tried it out extensively.

The next thing to do is to change macromedias flash-settings:

9.6 Flash-Settings

Go here.

And fight yourself through their nasty settings-manager. Set everything to ”0” or ”never allow”/”never ask again” and

delete all stored website-content. Give special attention to the ”webcam and mic”-options...

You might as well set the permissions of your .macromedia folder to *read only* - but that's kind of unnecessary because you want to make sure to edit the options mentioned above - to make sure that you don't allow websites to use your mic or webcam... (I actually take this one step further by disabling them in BIOS *and* sticking some neatly cut little piece of black cardboard on my webcam. *Just because you're paranoid doesn't mean they aren't after you*...) And if you set the parameters in the settings-manager accordingly nothing will be written to that folder anyway.

Now we come to the fun part. Finetuning Firefox using about:config. If you've never done this before: No reason to freak out. It's really easy.

9.7 about:config

(You can simply copy/paste these variables into the search-bar at the top: e.g. "browser.cache.disk.enable" and

then double-click on the entry that shows up to modify the settings.)

```
-$\,$-$\,$-disable browser cache: browser.cache.disk.enable:false
browser.cache.disk_cache_ssl:false browser.cache.offline.enable:false browser.
    cache.memory.enable:false browser.cache.disk.capacity:0 browser.cache.disk.
    smart_size.enabled:false browser.cache.disk.smart_size.first_run:false
    browser.cache.offline.capacity:0 dom.storage.default_quota:0 dom.storage.
    enabled:false dom.indexedDB.enabled:false dom.battery.enabled:false -$\,$-$
    \,$-disable history & localization browser.search.suggest.enabled:false
    browser.sessionstore.resume_from_crash:false geo.enabled:false -$\,$-$\,$-
    misc other tweaks: keyword.enabled:false network.dns.disablePrefetch:true ->
     very important when using TOR network.dns.disablePrefetchFromHTTPS -> very
    important when using TOR dom.disable_window_open_feature.menubar:true dom.
    disable_window_open_feature.personalbar:true dom.disable_window_open_feature
    .scrollbars:true dom.disable_window_open_feature.toolbar:true browser.
    identity.ssl_domain_display:1 browser.urlbar.autocomplete.enabled:false
    browser.urlbar.trimURL:false privacy.sanitize.sanitizeOnShutdown:true
    network.http.sendSecureXSiteReferrer:false network.http.spdy.enabled:false -
    $\,$-$\,$-> use http instead of google's spdy plugins.click_to_play:true -$
    \,$-$\,$-> also check each drop-down-menu under "preferences"->"content"
    security.enable_tls_session_tickets:false -$\,$-$\,$-> disable https-
    tracking security.ssl.enable_false_start:true -$\,$-$\,$-> disable https-
    tracking extensions.blocklist.enabled:false -$\,$-$\,$-> disble Mozilla's
    option to block/disable your addons remotely webgl.disabled:true -$\,$-$\,$
    -> disable WebGL ((url)\url{http://security.stackexchange.com/questions
    /13799/is-webgl-a-security-concern(/url))} network.websocket.enabled:false -
    $\,$-$\,$-> ***Tor Users: This is extremely important as it could blow your
    cover! See: (url)\url{http://pastebin.com/xajsbiyh***(/url)} -$\,$-$\,$-make
     your browsing faster: network.http.pipelining:true network.http.pipelining.
    ssl:true network.http.proxy.pipelining:true network.http.max-persistent-
    connections-per-proxy:10 network.http.max-persistent-connections-per-server
    :10 network.http.max-connections-per-server:15 network.http.pipelining.
```

```
maxrequests:15 network.http.redirection-limit:5 network.dns.disableIPv6:true
 network.http.fast-fallback-to-IPv4:false dom.popup_maximum Mine:10 network.
prefetch-next:false browser.backspace_action:0 browser.sessionstore.
max_tabs_undo:5 browser.sessionhistory.max_entries:5 browser.sessionstore.
max_windows_undo:1 browser.sessionstore.max_resumed_crashes:0 browser.
sessionhistory.max_total_viewers:0 browser.tabs.animate:0
```

(thanks to machinebacon for these last entries.

9.8 Prevent Browser Fingerprinting (still in about:config)

For all Firefox Versions after 17.0 (you should be using current versions and update them regularly anyway - to do this go to ”preferences”-¿”advanced”-¿”update” select: ”automatically install updates” & ”warn me if this will disable any of my addons”) (not required for iceweasel)

For the following changes right-click in about:config and select ”new”-¿”string” and enter in this order:

```
Variable: Value: general.useragent.override Mozilla/5.0 (Windows NT 6.1; rv
    :10.0) Gecko/20100101 Firefox/10.0 general.appname.override Netscape general
    .appversion.override 5.0 (Windows) general.oscpu.override Windows NT 6.1
    general.platform.override Win32 general.productSub.override 20100101 general
    .buildID.override 0 general.useragent.vendor (enter variable - but leave
    value blank) general.useragent.vendorSub (enter variable - but leave value
    blank) intl.accept_languages en-us,en;q=0.5 network.http.accept.default text
    /html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 network.http.
    accept-encoding gzip, deflate
```

This creates a fake-profile of your browser via the readable HTTP-headers it sends.

Check out if your browser is profilable.

With all the above settings I get 8.1 bits of identifying information at Panopticlick for my browser - which is *really* good.

Considering:

”In particular, a fingerprint that carries no more than 15-20 bits of identifying information will in almost all cases be sufficient to uniquely identify a particular browser, given its IP address, its subnet, or even just its Autonomous System Number.”

Source: EFF’s ”Browser Uniqueness” (page 3)

Also check your settings on ip-check.info - but don't rely on it. Apparently they are quite busy promoting their JonDonym-Browser and services - which quite frankly I don't think anyone needs. I would rather warn you to use it since according to this defcon-talk JAP/JonDonym has implemented tracking-features which are disabled by default but can be activated anytime. So don't use it.

Now, after having configured your host-based security and your web-browser we can start connecting to the web. But there are different options:

10 TOR (The Onion Router)

TOR is probably the most famous anonymizing-tool available. You could consider it a safe-web proxy. (**Update**: I wouldn't say that any longer. See the TOR-Warning below for more info.) Actually, simply put, it functions as a SOCKS-proxy which tunnels your traffic through an encrypted network of relays in which your ip-address can not be traced. When your traffic exits the network through so-called exit-nodes the server you are contacting will only be able to retrieve the ip-address of the exit-node. It's pretty useful - but also has a few drawbacks:

First of all it is slow as f**k. Secondly exit-nodes are often times honey-pots set up by cyber-criminals and intelligence agencies. Why? The traffic inside the TOR-network is encrypted - but in order to communicate with services on the "real" internet this traffic needs to be decrypted. And this happens at the exit-nodes - which are thus able to inspect your packets and read your traffic. Pretty uncool. But: you can somewhat protect yourself against this kind of stuff by *only* using SSL/https for confidential communications such as webmail, forums etc. Also, make sure that the SSL-certificates you use can be trusted, aren't broken and use secure algorithms. The above mentioned Calomel SSL Validation addon does a good job at this. Even better is the Qualys SSL Server Test.

The third bummer with TOR is that once you start using TOR in an area where it is not used that frequently which will be almost everywhere - your ISP will directly be able to identify you as a TOR user if he happens to use DPI (Deep Packet Inspection) or flags known TOR-relays. This of course isn't what we want. So we have to use a workaround. (For more info on this topic watch this vid: How the Internet sees you (27C3))

This workaround isn't very nice, I admit, but basically the only way possible to use TOR securely.

So, the sucker way to use TOR securely is to use obfuscated bridges. If you don't know what this is please consider reading the TOR project's info on bridges

Basically we are using TOR-relays which are not publicly known and on top of that we use a tool to hide our TOR-traffic and change the packets to *look like* XMPP-protocol.

Why does this suck? It sucks because this service is actually meant for people in real disaster-zones,

like China, Iran and other messed up places. This means, that everytime we connect to TOR using this technique *we steal bandwidth from those who really need it*. Of course this only applies if you live somewhere in the Western world. But we don't really know what information various agencies and who-knows-who collect and how this info will be used if, say, our democratic foundations crumble. You could view this approach as being *pro*active in the West whereas it is necessary and *re*active in the more unfortunate places around the world.

But, there is of course something we can do about this: first of all *only* use TOR when you have to. You don't need TOR for funny cat videos on youtube. Also it is good to have some regular traffic coming from your network and not only XMPP - for obvious reasons. So limit your TOR-use for when it is necessary.

The other thing you/we can do is set up our own bridges/relays and contribute to the network. Then we can stream the DuckTales the whole darn day using obfuscated bridges without bad feelings...

How to set up a TOR-connection over obfuscated bridges?

Simple: Go to -¿ The Tor project's special obfsproxy page and download the appropriate *pre-configured* Tor-Browser-Bundle.

Extract and run. (Though *never* as root!)

If you want to use the uber-secure webbrowser we configured above simply go to the TOR-Browsers settings and check the port it uses for proxying. (This will be a different port every time you start the TOR-Bundle.)

Then go into your browser and set up your proxy accordingly. Close the TOR-Browser and have phun! - But don't forget to: check if you're really connected to the network.

To make this process of switching proxies even more easy you can use the FireFox-addon: FoxyProxy. This will come in handy if you use a regular connection, TOR and I2P all through the same browser.

Tipp: While online with TOR using google can be quite impossible due to google blocking TOR-exit-nodes - but with a little help from HideMyAss! we can fix this problem. Simply use the HideMyAss! web interface to browse to google and do your searchin'. You could also use search engines like ixquick, duckduckgo etc. - but if you are up for some serious google hacking - only google will do... (Apparently there exists an alternative to the previously shut-down scroogle: privatelee which seems to support more sophisticated google search queries. I just tested it briefly after digging it up here. So you need to experiment with it.)

But remember that in case you do something that attracts the attention of some three-letter-organization HideMyAss! will give away the details of your connection. So, only use it in combination with TOR - *and*: don't do anything that attracts that kind of attention to begin with.

Warning: Using Flash whilst using TOR *can* reveal your real IP-Address. Bear this in mind! Also, double-check to have network.websocket.enabled set to false in your about:config! -¿ more info on that one here.

Another general thing about TOR: If you are really concerned about your anonymity you should *never* use anonymized services along non-anonymized services. (Example: Don't post on "frickkkin'-anon-ops-forum.anon" while browsing to your webmail "JonDoe@everybodyknowsmyname.com")

And BTW: For those who didn't know it - there are also the TOR hidden services...

One note of caution: When dealing with darknets such as TOR's hidden services, I2P and Freenet please be aware that there is some *really* nasty stuff going on there. In fact in some obscure place on these nets everything you can and can't imagine is taking place. This is basically a side-effect of these infrastructure's intended function: to facilitate an *uncensored* access to various online-services from consuming to presenting content. The projects maintaining these nets try their best to keep that kind of stuff off of the "official" search engines and indexes - but that basically is all that can be done. When everyone is anonymous - even criminals and you-name-it are.

To avoid that kind of exposure and thus keep your consciousness from being polluted with other people's sickness please be careful when navigating through these nets. Only use search-engines, indexes and trackers maintained by trusted individuals. Also, if you download anything from there make sure to *triple* check it with ClamAV. Don't open *even one* PDF-file from there without checking.

To check pdf-files for malicious code you can use wepawet. Or if you are interested in vivisecting the thing have a look at Didier Steven's PDFTools or PeePDF.

Change the file-ownership to a *user with restricted access* (i.e. *not* root) and set all the permissions to *read only*. Even better: only use such files in a machine. The weirdest code thrives on the darknets... I don't want to scare you away: These nets generally are a really cool place to hang out and when you exercise some common sense you shouldn't get into trouble.

(Another short notice to the **Germans**: Don't try to hand over stuff you may find there to the authorities, download or even make screenshots of it. This could get you into *serious* trouble. Sad but true. For more info watch this short vid.)

10.1 TOR-Warning

The above mentioned issues unfortunately aren't the only ones. I have come across more and more reasons *not* to use TOR:

- When using TOR you use about five times your normal bandwidth - which makes you stick out for your ISP - even with obfuscate bridges in use.

- TOR-nodes (!) and TOR-exit-nodes can be and are being used to deploy malicious code and to track and spy on users.

- There are various methods of de-anonymizing TOR-users: from DNS-leaks over browser-info-analysis to traffic-fingerprinting.

I won't explain all these issues in detail but if you are interested in finding out why TOR isn't safe to use (and you should if you actually think that TOR is making you anonymous) I recommend you watch these talks:

Attacking TOR at the Application-Layer

De-TOR-iorate Anonymity

Taking Control over the Tor Network

Dynamic Cryptographic Backdoors to take over the TOR Network

Security and Anonymity vulnerabilities in Tor

Anonymous Internet Communication done Right (I disagree with the speaker on Proxies, though. See info on proxies below.)

Owning Bad Guys and Mafia with Java-Script Botnets

And if you want to see how TOR-Exit-Node sniffing is done live you can have a look at this:

Tor: Exploiting the Weakest Link

To make something clear: I have nothing against the TOR-project. In fact I like it really much. But TOR is simply not yet able to cash in the promises it makes. Maybe in a few years time it will be able to defend against a lot of the issues that have been raised and illustrated. But until then **I can't safely recommend using it to anybody**. Sorry to disappoint you.

11 I2P

I2P is a so-called darknet. It functions differently from TOR and is considered to be way more secure. It uses a much better encryption and is generally faster. You can *theoretically* use it to browse the web - but it is generally not advised and even slower as TOR using it for this purpose. I2P has some cool sites to visit, an anonymous email-service and a built-in anonymous torrent-client.

For I2P to run on your system you need Open-JDK/JRE since I2P is a java-application. To install:

Go to-¿ The I2P's website download, verify the SHA256 and install:

```
$ cd /directory/you/downloaded/the/file/to && java -jar i2pinstall_0.9.4.jar
```

Don't install as root - and even more important: *Never* run as root!

```
To start:
$ cd /yourI2P/folder ./i2prouter start To stop:
$ cd /yourI2P/folder ./i2prouter stop
```

Once running you will be directed to your Router-Console in FireFox. From there you have various options. You should consider to give I2P more bandwidth than default for a faster and more anonymous browsing experience.

The necessary browser configuration can be found here.

For further info go to the project's website.

12 Freenet

A darknet I have not yet tested myself, since I only use TOR and I2P is Freenet. I heard that it is not that populated and that it is mainly used for filesharing. A lot of nasty stuff also seems to be going on on Freenet - but this is only what I heard and read about it. The nasty stuff issue of course is also true for TOR's hidden services and I2P. But since I haven't been on it yet I can't say anything about that. Maybe another user who knows Freenet better can add her/his review.

Anyhow...:

You get the required software here.

If you want to find out how to use it - consult their helpsite.

13 Secure Peer-to-Peer-Networks

- GNUNet
- RetroShare

14 Mesh-Networks

If you're asking yourself what mesh-networks are take a look at this short video.

- guifi.net
- Netsukuku Community

- OpenWireless
- Commotion
- FabFi
- Mesh Networks Research Group
- Byzantium live Linux distro for mesh networking

Thanks to cyberhood!

15 Proxies

I have not yet written anything about proxy-servers. In short: Don't *ever* use them.

There is a long and a short explanation. The short one can be summarized as follows:

- Proxy-servers often sent xheaders containing your actual IP-address. The service you are then communication to will receive a header looking like this:

```
X-Forwarded-For: client, proxy1, proxy2
```

This will tell the server you are connecting to that you are connecting to him via a proxy which is fetching data on behalf of... you!

- Proxy servers are infested with malware - which will turn your machine into a zombie within a botnet - snooping out all your critical login data for email, banks and you name it.

- Proxy servers can read - and *modify* - all your traffic. When skilled enough sometimes even circumventing SSL.

- Proxy servers can track you.

- Most proxy servers are run by either criminals or intelligence agencies.

Seriously. I really recommend watching this (very entertaining) Defcon-talk dealing with this topic. To see how easy e.g. java-script-injections can be done have a look at beef.

16 VPN (Virtual Private Network)

You probably have read the sections on TOR and proxy-servers (do it now - if you haven't) and now you are asking yourself: "Fuck, what can I use to browse the web safely and anonymously????"

Well, there is a pretty simple solution. But it will cost you a few nickels. You have to buy a premium-VPN-service with a trustworthy VPN-provider.

If you don't know what a VPN is or how it works - check out this video.

Still not convinced? Then read what lifehacker has to say about it.

Once you've decided that you actually want to use a VPN you need to find a trustworthy provider. Go here to get started with that.

Only use services that offer OpenVPN. Basically all the other protocols aren't that secure. Or at least they can't compare to OpenVPN.

Choose the most trustworthy service you find out there and be paranoid about it.

A trustworthy service doesn't keep logs. If you choose a VPN, read the complete FAQ, their Privacy Policy *and* the Terms of Service. Check where they're located and check local privacy laws. And: Don't tell people on the internet which service you are using.

You can get yourself a second VPN account with a different provider you access through a VM. That way VPN#1 only knows your IP-address but not the content of your communication and VPN#2 knows the content but not your IP-address.

Don't try to use a free VPN. Remember: If you're not paing for it - *you* are the product.

17 The Web

If for some unimaginable reason you want to use the ”real” internet - you now are equipped with a configuration which will hopefully make this a much more secure endeavour. But still: Browsing the internet and downloading stuff is the greatest vulnerability to a linux-machine. So use some common sense.

17.1 RSS-Feeds

Please be aware that using RSS-feeds can be used to track you and the information-sources you are using. Often RSS-feeds are managed through 3rd-party providers and not the by the original service you are using.

Web-bugs are commonly used in RSS-tracking. Also your IP-address and other available browser-info will be recorded.

Even when you use a text-based desktop-feedreader such as newsbeuter - which mitigates tracking though web-bugs and redirects - you still leave your IP-address.

To circumvent that you would want to use a VPN or TOR when fetching your RSS-updates.

If you want to learn more about RSS-tracking read this article.

18 Secure Mail-Providers:

Please consider using a secure email-provider and encourage your friends and contacts to do the same. All your anonymization is worthless when you communicate confidential information in an unencrypted way with someone who is using gmx, gmail or any other crappy provider. (This also applies if you're contemplating setting up your own mail-server.)

If possible, encrypt everything, but especially confidential stuff, using gpg/enigmail.

- lavabit.com (SSL, SMTP, POP)
- hushmail.com (SSL, SMTP, **no** POP/IMAP - only in commercial upgrade)
- vfemail.net (SSL, SMTP, POP)

I found these to be the best. But I may have missed others in the process.

Hushmail also has the nice feature to encrypt "inhouse"-mails, i.e. mail sent from one hushmail-account to another. So, no need for gpg or other fancy stuff.

The user cyberhood mentioned these mail-providers in the other **#!** thread on security.

- autistici.org (SSL, SMTP, IMAP, POP) Looks alright. Maybe someone has tested it already?
- mailoo.org (SSL, SMTP, IMAP, POP) Although I generally don't trust services that can not present themselves without typos and grammatical errors - I give them

  the benefit of the doubt for they obviously are *French*. Well, you know how the French deal with foreign languages...
- countermail.com (SSL, SMTP, IMAP, POP) See this review
- https://help.riseup.net/en/about-usriseup.org

  You need to prove that you are some kind of activist-type to get an account with them. So I didn't bother to check out their security. This is how they present themselves:RISEUP WROTE:

  > The Riseup Collective is an autonomous body based in Seattle with collective members world wide. Our purpose is to aid in the creation of a free society, a world with freedom from want and freedom of expression, a world without oppression or hierarchy, where power is shared equally. We do this by providing communication and computer

resources to allies engaged in struggles against capitalism and other forms of oppression.

Edit: I changed my mind and will not comment on Riseup. It will have its use for some people and as this is a technical manual I edited out my political criticism to keep it that way.

19 Disposable Mail-Addresses

Sometimes you need to register for a service and don't want to hand out your real mail-address. Setting up a new one also is a nuisance. That's where disposable mail-addresses come in. There is a firefox-addon named Bloody Vikings that automatically generates them for you. If you rather want to do that manually you can use some of these providers:

- anonbox
- anonymouse/anonemail
- trash-mail
- 10 Minute Mail
- dispostable
- SilentSender
- Mailinator

It happens that websites don't allow you to register with certain disposable mail-addresses. In that case you need to test out different ones. I have not yet encountered a site where I could not use one of the many one-time-address out there...

20 Secure Instant-Messaging/VoIP

Using Skype is not advised from a security standpoint. Although Skype communication is encrypted there are a few ways to attack it. Also, you probably don't want to *trust* Skype to keep all your data safe, do you?

Instead you can use:

20.1 TorChat

To install:

```
$ sudo apt-get install torchat
```

TorChat is generally considered to be *really* safe - employing end-to-end encryption via the TOR network. It is both anonymous and encrypted.

Obviously you need TOR for it to function properly.

you find instructions on how to use it.

20.2 OTR (Off-the-Record Messaging)

OTR is also very secure. Afaik it is encrypted though not anonymous.

Clients with native OTR support:

Jitsi

Climm

Clients with OTR support through Plugins:

Pidgin

Kopete

XMPP generally supports OTR.

Here you find a tutorial on how to use OTR with Pidgin.

20.3 Secure and Encrypted VoIP

As mentioned before - using Skype is not advised. There is a much better solution:

Jitsi

Jitsi is a chat/VoIP-client that can be used with different services, most importantly with XMPP. Jitsi doesn't just offer **chat**, **chat with OTR**, **VoIP-calls over XMPP**, **VoIP-video-calls via XMPP** - but also the ZRTP-protocol, which was developed by the developer of PGP, Phil Zimmerman.

ZRTP allows you to make fully end-to-end encrypted *video-calls*. Ain't that sweet?

If you want to know how that technology works, check out these talks by Phil Zimmerman at Defcon. (Defcon 15 — Defcon 16)

Setting up Jitsi is pretty straightforward.

Here is a very nice video-tutorial on how get started with Jitsi.

21 Social Networking

21.1 Facebook

Although I actually don't think I need to add this here - I suspect other people coming to this forum from google might need to consider this: *Don't use Facebook*!

Apart from security issues, malware and viruses Facebook itself collects every bit of data you hand out: to store it, to sell it, to give it to the authorities. And if that's still not enough for you to cut that crap you might want to watch this video.

And *no*: Not using your real name on Facebook isn't helping you anything. Who are your friends on Facebook? Do you always use an IP-anonymization-service to login to Facebook? From where do you login to Facebook? Do you accept cookies? LSO-cookies? Do you use SSL to connect to Facebook? To whom are you writing messages on Facebook? What do you write there? Which favorite (movies — books — bands — places — brands)-lists did you provide to Facebook which only need to be synced with google-, youtube-, and amazon-searches to match your profile? Don't you think such a massive entity as Facebook is able to connect the dots? You might want to check out this vid to find out how much Facebook actually *does* know about you. Still not convinced? (Those who understand German might want to hear what the head of the German Police Union (GDP), Bernhard Witthaut, says about Facebook on National TV...)

For all of you who *still* need more proof regarding the dangers of Facebook and mainstream social media in general - there is a defcon-presentation which I urge you to watch. *Seriously*. Watch it.

Well, and then there's of course Wikipedia's collection of criticism of Facebook. I mean, come on.

21.2 Alternatives to Facebook

Friendica is an alternative to Facebook recommended by the Free Software Foundation

Lorea seems a bit esoteric to me. Honestly, I haven't wrapped my head around it yet. Check out their description:LOREA WROTE:

> Lorea is a project to create secure social cybernetic systems, in which a network of humans will become simultaneously represented on a virtual shared world.
>
> Its aim is to create a distributed and federated nodal organization of entities with no geophysical territory, interlacing their multiple relationships through binary codes and languages.

Diaspora - but there are some doubts - or I'd better say: *questions* regarding diasporas security.

But it is *certainly* a better choice than Facebook.

One last thing:

22 Passwords

Always make sure to use good passwords.

To generate secure passwords you can use:

22.1 pwgen

Installation:

```
$ sudo apt-get install pwgen
```

Usage:

```
pwgen ( OPTIONS ) ( pw_length ) ( num_pw ) Options supported by pwgen: -c or -$
    \,$-capitalize Include at least one capital letter in the password -A or -$
    \,$-no-capitalize Don't include capital letters in the password -n or -$\,$-
    numerals Include at least one number in the password -0 or -$\,$-no-numerals
     Don't include numbers in the password -y or -$\,$-symbols Include at least
    one special symbol in the password -s or -$\,$-secure Generate completely
    random passwords -B or -$\,$-ambiguous Don't include ambiguous characters in
     the password -h or -$\,$-help Print a help message -H or -$\,$-sha1=path/to
    /file(#seed) Use sha1 hash of given file as a (not so) random generator -C
    Print the generated passwords in columns -1 Don't print the generated
    passwords in columns -v or -$\,$-no-vowels Do not use any vowels so as to
    avoid accidental nasty words
```

Example:

```
$ pwgen 24 -y
```

Pwgen will now give you a list of password with 24 digits using at least one special character.

To test the strength of your passwords I recommend using Passfault. *But*: Since Passfaults' symmetric cypher is rather weak I advise not to use your real password. It is better to substitute each character by another similar one. So you can test the strength of the password without

transmitting it in an insecure way over the internet.

If you have reason to assume that the machine you are using is compromised and has a keylogger installed you should generally only use virtual keyboards to submit critical data. They are built in to every OS afaik.

Another thing you can do is use:

22.2 KeePass

KeePass stores all kinds of password in an AES/Twofish encrypted database and is thus highly secure and a convenient way to manage your passwords.

To install:

```
$ sudo apt-get install keepass2
```

A guide on how to use it can be found here.

22.3 Live-CDs and VM-Images that focus on security and anonymity

Tails Linux The classic. Debian-based.

Libert Linux Similar to Tails. Gentoo-based.

Privatix Live-System Debian-based.

Tinhat Gentoo-based.

Pentoo Gentoo-based. Hardened kernel.

Janus VM - forces all network traffic through TOR

23 Further Info/Tools:

TOR

I2P

Securing Debian Manual

Electronic Frontier Foundation

EFF's Surveillance Self-Defense Guide

Schneier on Security

Irongeek

SpywareWarrior

SecurityFocus

Wilders Security Forums

Insecure.org

CCC (en)

Eli the Computer Guy on Security

Digital Anti-Repression Workshop

The Hacker News

Anonymous on the Internets!

#! Privacy and Security Thread (Attention: There are some dubious addons listed! See my post there for further info.)

EFF's Panopticlick

GRC

Rapid7 UPnP Vulnerability Scan

HideMyAss! Web interface

Browserspy

ip-check.info

IP Lookup

BrowserLeaks

Whoer

evercookie

Sophos Virus DB

f-secure Virus DB

Offensive Security Exploit DB

Passfault

PwdHash

Qualys SSL Server Test

MyShadow

Security-in-a-Box

Calyx Institute

CryptoParty

Self-D0xing

Wepawet

23.1 German only:

awxcnx

anondat

SemperVideo

SemperVideo (youtube)

Fefes Blog

heise

golem

CCC (de)

FoeBud

German Privacy Foundation

23.2 Postscript:

If you find any error in this guide please don't hesitate to reply with an explanation. Also, if you have anything to add please also use the reply function. Since this is my first "real" post on the **#!** forums I don't know how long the edit-function is available for regular posts. Should it be usable indefinitely I will edit this original post to include all the additional information you will provide. This way we keep all the required info in one place. Thanks!

...and keep sorcering!

(Edit: Apparently I can edit the hell out of this post. So I will be constantly updating this guide in the future. I already scrambled together all the info I found noteworthy from the **#!** Privacy and Security Thread. So you should *in theory* find everything you need from there in this manual, too. But you know how personal opinions differ. So please raise your hand if you find I missed something.

I will also work on migrating this guide into the **#!**-wiki in the future.)