

Forum:[Hack Forums](#)

Thread: [\[TUT\] \[KALI\] How to build a Fed-proof Hacking Device](#)

Nomad.Null - December 23, 2015, 5:30 am

The Guide to Hack-Box Encryption, Security, and Counter-Forensics

Greetings, readers. We the members of Legends would like to present this tutorial as a gift to the community in response to recent events. Given the social and political issues going on in the world with the debate of Government Surveillance increasing in the United States, the increased presence of cyber investigators and honeypots on the internet, and the general crackdown being spearheaded by Interpol, NCA, NSA, and FBI, we felt that now was an appropriate time to address the issues of counter forensics and hackbox security.

We have noticed a disturbing number of members that are either ignorant, misinformed, or overly-confident about computer forensics, and what happens if the federales bust in your door and rip your computer out of the wall to dissect it in a sterile forensic lab. We have partnered together to deliver the knowledge you need to secure your computer system and simplified it such that the average user can very easily and cheaply harden your computer system well beyond the abilities of even the most skilled Forensic Analysts.

This guide covers, in detail, the process in simple, easy-to-follow steps for completely encrypting your hard drive and setting up counter-intrusion measures that make physically breaking into your system all but an impossibility. If you are concerned about your privacy, your physical security, worries of theft, or seizure of your devices, this guide will help you.

Good luck and thank you for reading,

- Armada and True Demon, The Legends Group

Securing your KALI Device

Greetings, Kali-newbies

This is True Demon here to explain a few not-so-secret nuances of the Kali-linux Pentesting distribution. There have been a number of events recently on the forums and elsewhere that have led me to believe some of you are a bit terrified of what would happen to you if the feds busted down your door, put you in cuffs, and sent your hard drive to a lab to be ripped apart and scrutinized for evidence.

Well, today, I am going to teach you about Anti-Forensics for the purpose of hardening and protecting your hard-drive from being analyzed by unwanted parties.

What is Anti-Forensics?

When an investigation is being conducted, there is a process for gathering evidence, analyzing that evidence, and the resulting conclusions that come from that analysis that lead investigators to make arrests. This is what "Forensics" entails. Anti-forensics is the opposite, where you are trying to conceal, subvert, or destroy that evidence. This particular aspect of Anti-Forensics that I am going to teach you about is Hard drive Encryption specifically for the Kali-Linux operating system.

The reason I am covering Kali specifically is because it is extremely popular in the pentesting

community, and this tutorial will also apply to any other linux distribution that is compatible with LUKS encryption.

There are two aspects to this guide: Creating a mobile pentesting platform on a USB Flashdrive and encrypting that platform to make it impossible to read without the passkey. This ensures that even if your hard drive is stolen, seized, or compromised, it cannot be unlocked unless you are there to physically enter the password. This protects your information from falling into the wrong hands. This is especially useful for those pentesters whose hard drives contain a lot of confidential information on their hard drive—of course this applies to anybody with Kali who has something to hide though.

Creating your Bootable Linux USB

Something that I am utterly shocked at is how few people create mobile pentesting platforms these days. It is extremely easy, highly secure, and it allows you to create a mobile pentesting platform that you can literally carry in your pocket and destroy on a whim. This can be done with any USB Flash drive over 8GB in size, though I usually recommend at least a 32GB USB 3.0 drive, and nothing less. I personally use a 128GB USB3.0 drive that I picked up on sale at a BestBuy for \$50. You could also put a SSD in a 2.5" SATAIII drive enclosure, or whatever it is you like to do if you have more specialized hardware. Hell, you can even do this with a Raspberry Pi!

To start, you want to grab your Kali .iso from www.kali.org/downloads/

I highly recommend Kali 2.0. It runs like a dream, in my humble opinion, and the GNOME environment looks just beautiful. As far as linux distributions go, it is one of the prettiest and most user-friendly I have seen since Ubuntu was released.

If you have Kali burned onto a DVD and want to create a Kali persistent USB drive via Kali Live, then skip this next section. If you are using Windows to do this, you'll want to keep reading the section below.

Installing Kali Persistence on a USB Flashdrive via Windows

This is a guide on how to set up a Kali-linux Persistent USB drive using Windows and windows-only applications. I did not create this guide. It was created by the user "Kitten" on Null-byte.wonderhowto.com. All thanks for this section go to them.

<http://null-byte.wonderhowto.com/how-to/install-kali-live-usb-drive-with-persistence-optional-0162253/>

Creating a Bootable Kali USB Flashdrive via Linux or Kali Live

Now, assuming you have Kali installed as your primary OS, you want to do things the interesting way with Linux, or you just plain hate window's fucking guts, you can pop in your Kali Live CD or boot up from Linux and create a bootable USB drive this way, which I consider vastly easier.

Boot into Linux/Kali as root or sudo to root and plug in your USB flash drive you intend to make bootable.

Start by running an fdisk -l to list your running hard drives.

The disk /dev/sdb is the one I want to write to. (sda is my windows NTFS hard drive, so ignore it.)

Now, change directories to wherever you have your kali.iso file located at. I moved it to my root/desktop beforehand, so...

```
cd /root/Desktop
```

MAKE SURE THAT YOU HAVE CORRECTLY IDENTIFIED THE USB DRIVE AS /DEV/SDx BECAUSE THE NEXT COMMAND WILL FORMAT EVERYTHING ON THAT DISK!!!

Use the dd command to image your Kali .iso onto the USB drive. Your input will be if=kali-linux-2.0-amd64 or whatever your kali distribution is, and your output will be of=/dev/sdx with a blocksize of bs=512k. This is considered a pretty standard and stable blocksize, so it is what I recommend. Going any higher can lead to corruption, and you may lose space in place of that little performance boost you'd be hoping for.

So, the command on my system looks like

```
Code dd if=kali-linux-2.0-amd64.iso of=/dev/sdb bs=512k
```

The dd command does NOT provide you any feedback, so don't be scared if it takes it a while to image the file to the flash drive. The largest kali image to-date is only about 3.5gb in size, so it can take up to 30 minutes if you have a really slow computer.

Once it is done imaging, you will see the feedback.

Do another sudo fdisk -l and take a look at your designated device. If it was sdb, it should look very similar to my setup.

Go ahead and remove the Kali Live CD and shutdown. We are now going to set up the persistence module.

Partition your USB Flashdrive

When you boot up into your USB drive, you will have a choice to boot into the Live environment, Live USB Persistence, or Live USB Encrypted Persistence. If you want to be unprotected, by all means, go with the standard persistence, but if you came here to learn how to encrypt your Kali Live USB disk, you are going to want to boot with Live USB Encrypted Persistence, so go ahead and do that.

First, we need to create a new partition, so we're going to write a quick line of code to create a partition for us to write the LUKS encryption to. I am going to end mine at 125gb, since that is where the useable amount of space ends on my flash drive. If you are using an 8gb flash drive, you might want to end at 7 gb, or if you're using a 32gb flashdrive, you would want to end at 29 or 30gb. So adjust accordingly. You can always resize your partition with gparted later. Write the following lines of code to partition your drive with the appropriate size.

```
Code end=125gb
```

```
read start _ < <(du -bcm kali-linux-2.0-amd64.iso | tail -1); echo $start  
parted /dev/sdb mkpart primary $start $end
```

The kernel will likely recommend a different start-point for the partition. Accept the suggestion to prevent parted from overwriting the partition. It may also warn you that the partition is not aligned for best performance, but you can ignore this. You will not need to update /etc/fstab as it suggests, since Kali will do this for us once we create the persistence module.

Encrypt your drive!

Now, we are going to setup the encryption with LUKS. If you don't want encryption, skip this step.

This will work on any hard drive or USB flashdrive. So if you already have Kali installed on your HDD, you can do this the same way. Just ensure you are encrypting the correct hard drive and partition.

You will be asked to create a passphrase when you do this. **DO NOT FORGET YOUR PASSPHRASE OR YOU WILL LOSE EVERYTHING!** I highly recommend a strong password no less than 10 characters in full key-space. That means upper&lowercase characters, numbers, and symbols. For example GO jump off a 100' CLIFF! is a pretty strong and acceptable password, in my opinion. ?

```
Code cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb3
```

It will take just a minute or two to set up the encryption on the partition. It was too easy, right?

Now we will open the luks channel to the hard drive and setup our persistence module.

```
Code cryptsetup luksOpen /dev/sdb3 my_usb_label
```

Next, format sdb3 into an ext4 filesystem and label it "persistence" for brevity.

```
Code mkfs.ext4 -L persistence /dev/mapper/my_usb_label  
e2label /dev/mapper/my_usb_label persistence
```

Well done! Now we are going to mount the encrypted partition and setup the persistence configuration file so we can keep all of our files between reboots, then unmounts it again.

```
Code mkdir -p /mnt/my_usb_label  
mount /dev/mapper/my_usb_label /mnt/my_usb_label  
echo "/ union" > /mnt/my_usb_label/persistence.conf  
umount /dev/mapper/my_usb_label
```

That's it! You're done! All you have to do is reboot and log into the Encrypted Persistent Live USB again to see what changes have been made.

This next one is an optional step. You should only view and follow these instructions for paranoia-levels of security and information assurance.

Spoiler ([Click to View](#))

Optional: Set up LUKS Nuke

Now, this next step is optional, but I HIGHLY recommend it, especially if you do NOT want anything on that hard drive to be found by anyone no matter what. I will show you how to set up an

Auto-Nuke module, so that if your hard drive falls into the wrong hands and someone attempts to brute-force the key or you, yourself want to nuke the drive on a whim, you can.

Code `cryptsetup luksAddNuke /dev/sdb3`

For this, you will want to set up a short, easy password to remember that is very weak. Why would you want this? Because obviously if you care about your information that much that you would rather see it destroyed than fall into the wrong hands, your worst nightmare is someone who wants to brute-force your usb drive. LuksNuke will delete the keyring, leaving the information encrypted, rendering inaccessible. It is the equivalent to locking the door and throwing away the key (in the fiery depths of hell). Set it to something like “nukeme” or “admin” or something a brute-force program would guess right away.

You can also setup a backup file on a separate hard-drive so that, if you do, in fact, nuke everything, you can get it all back by leaving a secret backup of your keys in another location, like buried two meters under your house foundation...too paranoid? Don't kid yourself... Anyway, moving on...

Change directories to another hard drive or flash drive. This will be the location where the file is saved to, so make sure it is safe, secure, and secret.

```
cd
/media/Super_Secret_Encrypted_MicroSD_card/Super_Secret_Filepath/Super_Secret_Encrypted_Folder
```

Code `cryptsetup luksHeaderBackup --header-backup-file luksheader.back /dev/sdb3`

Now check the file's contents with `>> file luksheader.back`

Now, encrypt this file with openssl encryption so that we can only access it with a DIFFERENT password than either of these. I would recommend something even stronger than your hard drive password itself.

Code `openssl enc -aes-256-cbc -salt -in luksheader.back -out luksheader.back.enc`

I would then recommend deleting the unencrypted backup file. That way, anyone trying to read the file will encounter...this...

YOU MUST REMEMBER THE NAME OR IT WILL BE IRRETRIEVABLE!!!

If you feel like experimenting, you can nuke your hard drive by rebooting and nuking the drive by entering your “nuke” password. You won't be able to get into the encrypted persistence drive anymore, so you will have to boot into Live Kali without persistence or encrypted persistence. You can check to see what happened to your keyslots by typing:

Code `cryptsetup luksDump /dev/sdb3`

This will dump the header for sdb3. You'll likely see every keyslot reading as empty. This means the nuke worked. Now we just have to get it back ?

First, we need to decrypt the file that we encrypted earlier.

Code `Openssl enc -d -aes-256-cbc -in luksheader.back.enc -out luksheader.back`

Enter your secret passphrase:

Now we just have to restore the header!

Code cryptsetup luksHeaderRestore --header-backup-file luksheader.back /dev/sdb3

It will warn you that this will overwrite the existing header. Since there are no keys there, obviously you can't do any worse damage, so go ahead and answer YES.

Now do another cryptsetup luksDump /dev/sdb3 and you should see a proper dumpfile that shows existing keys, one for unlocking the drive, and another for nuking it!

Reboot and enter your encrypted persistence drive! Congratulations, you now have a fully nuke-able, FBI-proof hard drive!

Hopefully this guide will help you feel a bit safer about using a bootable pentesting drive and not worry so much about what you would do if someone stole your computer/hard drive. You can now continue your life totally worry-free about what secrets lay dormant in your Kali distro.

- Your Friendly Neighborhood Demon ?

Gate-7-Hools - December 23, 2015, 8:40 am

First one to comment? Anyway Great and informative tut there mate! Keep it up!

Chainfire92 - December 23, 2015, 2:26 pm

thank you very much for this Tutorial very helpful and informative ?

EDIT:

can you maybe do a video Tut about this? :)

Nobilma - December 23, 2015, 3:53 pm

Wow, lots of good information.

The Defiant One - December 24, 2015, 7:07 pm

Very nice tutorial. Thanks!

TrooperJimmy - December 26, 2015, 12:23 am

This is the best thread I've came across on this forum since the day I joined, great job!

Naidoxe - December 26, 2015, 2:56 am

Well now I can feel a bit more comfortable when I come in contact with my feds :D

Light - December 26, 2015, 10:07 am

I've seen your around a lot in the hacking section, and you always seem to be posting high quality content, once again you did it. Nice job and great tutorial.

Controlled - December 27, 2015, 6:55 pm

with how many views this has with so little replies is disappointing , this is a really HQ post. Thanks for sharing this bro. I love how detailed this TUT is, plus its really useful!

Triads - January 4, 2016, 4:14 am

Cant use this because of my OS , Very HQ thread!

Thanks for putting effort in securing other peoples PC's

kamrulahsan06 - January 7, 2016, 9:15 pm

Very HQ tutorial. Thanks for your effort :)

BIGKING - January 9, 2016, 6:06 pm

Time to reinstall in kali and I only installed it first time ever earlier lmao thanks a lot !!

gimi - January 10, 2016, 8:57 am

nice tut thank you :)

Mr. Corleone - January 10, 2016, 10:21 am

Excellent tutorial. I really appreciate that you share your knowledge on this topic with the community ;-)

I've learned more than I thought, now I can hide my por- I mean 31337 hacks ^^

CryPTer006 - January 11, 2016, 5:48 am

I am using Kali for very long time and i say kali is the best hacking operating system.

Mr. Robot - January 11, 2016, 5:50 am

How am I part of this amazing group ;/

One of the few HQ members I haven't stalked yet - Your on my list, let me find the time you god like creature.

darma404 - January 11, 2016, 7:08 am

nice HQ Tutorial , thanks for share

FRANKIE - January 11, 2016, 7:12 am

Thread deserves a sticky.

gr0p0pl - March 22, 2016, 5:20 am

Hi everyone,

I am using Kali with 'Live USB Encrypted Persistence', however my 'encrypted partition' (4.018GB) is now too small.

I tried to increase the encrypted partition size by following this -

help.ubuntu.com/community/ResizeEncryptedPartitions

but there is emphasis on 'reducing the size' and not 'increasing the size', so i ended up getting very confused.

Can anyone help me do this in a more user friendly way or 'how to' please :)

Pekls - March 22, 2016, 8:39 am

lets go download Linux asap ^^

mixnet - March 22, 2016, 11:20 am

Would you say there is a security advantage to using your method versus using an encrypted Kali VM (via VMWare)?

[Except for the it being easier to dispose of as you could destroy the USB.]

Nomad.Null - March 22, 2016, 5:20 pm

(Mar 22, 2016 11:20 AM)mixnet Wrote: [\[CITE\]](#) Would you say there is a security advantage to using your method versus using an encrypted Kali VM (via VMWare)?

[Except for the it being easier to dispose of as you could destroy the USB.]

The encryption is the same either way, so really there is no other disadvantage than it being easier to destroy a VM and completely eradicate the data with a powerful erasure tool. Throwing your USB in a river can be just as effective though.

Honestly, using the USB has greater advantages because you use the full potential of your system's hardware as opposed to a VM which splits your resources between two operating systems.

Either way, you can encrypt the data without worry of any weaknesses in play. It is the same in both cases, provided you are using LVM LUKS encryption.

Toucan - March 22, 2016, 7:29 pm

Thanks for the high quality tutorial!

s3lkies - March 22, 2016, 7:38 pm

Im having problems partitioning the usb , i follow the steps however i get cannot access 'kali-linux-2.0-amd-64.iso': no such file or directory

The kali linux iso i installed (newest version) has a different name , i tried that and still got the same problem , any ideas ?

Great tut write up though, hopefully i can get it working!

mixnet - March 23, 2016, 1:59 am

(Mar 22, 2016 05:20 PM)True Demon Wrote: [\[CITE\]](#) The encryption is the same either way, so really there is no other disadvantage than it being easier to destroy a VM and completely eradicate the data with a powerful erasure tool. Throwing your USB in a river can be just as effective though.

Honestly, using the USB has greater advantages because you use the full potential of your system's hardware as opposed to a VM which splits your resources between two operating systems.

Either way, you can encrypt the data without worry of any weaknesses in play. It is the same in both cases, provided you are using LVM LUKS encryption.

Thanks :)

Ozedath - March 23, 2016, 1:07 pm

Code dd if=kali-linux-2.0-amd64.iso of=/dev/sdb bs=512k

I think the block size only dictates how fast the operation is being performed; the larger the block size the faster the write (the only thing being different is the padding at the end of the image once copied to the USB (it must be a multiple of block size in size)) since the operation is done sequentially in chunks of block size.

I haven't really worked with the USB version of Kali and was wondering what happens if you are pentesting and suddenly have to unplug the USB? You essentially have two partitions, no? One encrypted that is used to store the image while the Kali is offline and one in plaintext used during the pentest, meaning that when unplugged the plaintext version is stored as is (unencrypted) onto its respective partition, no?

Nomad.Null - March 23, 2016, 6:05 pm

(Mar 23, 2016 01:07 PM)Ozedath Wrote: [\[CITE\]](#) Code dd if=kali-linux-2.0-amd64.iso of=/dev/sdb bs=512k

I think the block size only dictates how fast the operation is being performed; the larger the block size the faster the write (the only thing being different is the padding at the end of the image once

copied to the USB (it must be a multiple of block size in size)) since the operation is done sequentially in chunks of block size.

I haven't really worked with the USB version of Kali and was wondering what happens if you are pentesting and suddenly have to unplug the USB? You essentially have two partitions, no? One encrypted that is used to store the image while the Kali is offline and one in plaintext used during the pentest, meaning that when unplugged the plaintext version is stored as is (unencrypted) onto its respective partition, no? That's an interesting note on the hard drive writing process. I still believe that the file system understands and stores data in segmented blocks, despite being in a solid state, but I would have to assume you are correct about the write speed. Whether or not it remains stored in 512k chunks, I can't be sure. It's been a LONG time since I took an A+ exam, lol.

As for unplugging the USB while it's plugged in...

Actually no, it's more like ripping your flashdrive out of a usb port in the middle of a read/write operation. It can cause data corruption and file system errors, especially if you are using encrypted persistence. The data is in a persistent encrypted state. That means that it doesn't become decrypted just because you enter your password. In fact, it never can be decrypted again without destroying the files and formatting the partition. Your key only makes the data readable by transforming it from ciphertext to plaintext before being viewed.

I've had to do it a couple of times, and it never really caused issues for me, but it's not healthy for the drive or for the OS. If it's doing anything important, especially something like updates or installing critical software, or any other interaction with the kernel, you can easily cause a fatal error. It's best avoided if you can help it. You are right though, it is just a second partition that acts like an additional hard drive whenever activated. You should treat it like a non-hotswappable hard drive. ?