Live-build :

[https://arpinux.developpez.com/construire-un-live-debian/#](https://arpinux.developpez.com/construire-un-live-debian/)

Awesome pentest : <https://github.com/enaqx/awesome-pentest#ctf-tools>

Kali tools : <https://tools.kali.org/tools-listing>

[https://live-team.pages.debian.net/live-manual/html/live-manual/](https://live-team.pages.debian.net/live-manual/html/live-manual/customizing-binary.en.html)

1. With the TEST :

./build.sh --distribution kali-rolling --verbose --variant custom -- \ --apt-http-proxy=${http\_proxy}

1. -- --variant custom
   * Packages needed : kali-desktop-live, kali-linux (check with test custom 3)
     + Packages-background : xfce … if wanted
   * Packages added : file in package-lists/\*.list.chroot
   * Add groups :
     + in includes.chroot/usr/lib/live/config/user-setup : LIVE\_USER\_DEFAULT\_GROUPS="audio cdrom dip floppy video plugdev netdev powerdev scanner bluetooth fuse"
   * Auto/config :
     + Langage :
     + Create new user :
   * Add file :
     + In all user : /includes.chroot/etc/skel : like .bashrc
     + In specific usr : create the path and add the file in
     + In etc : create the path in includes.chroot and add the file : includes.chroot/etc/var/www/index.html 🡪 create path both in etc/var and /var/ weird
   * Add or edit config or libraries :
     + In hooks/live : echo hola > /root/test.txt
     + Edit config service like ssh : in kali-config/custom/hooks/live/test.hook.chroot
   * Ex to config sshd : ex : echo 'systemctl enable ssh' >>  kali-config/common/hooks/01-start-ssh.chroot

* Create service in kali-config/common/includes.chroot/lib/systemd/name.service
* Create new user :
  + apt-get source debian-installer
  + Mkdir -p config/kali-user
  + Cp ../debian-installer-\*/build/preseed.cfg config/debian-installer/
  + Sed -i ‘s/make-user boolean false/make-user boolean true/’ config/debian-installer/preseed.cfg
  + Echo « d-i passwd/root-login boolean false » >> config/debian-installer/preseed.cfg

1. To remember :

* Binary/isolinux : dossier de config du bootloader syslinux.
* Binary/live : dossier du live proprement dit qui contient le SquashFS (système de file compressé en lecture seule sous linux) et l’image de démarrage initrd.img, ainsi que le vmlinuz appelé par le bootloader
* Cache : cache du live-build avec les paquets nécessaires a la constructio ainsi que le bootstrap de base

1. Documentation :

* Bootstrap : barebone of the foundation on which the customized ISO would be buiilt n, it provide all the files needed to create the ISO
* Chroot : creating a special directory that isolates applications from the rest of the system.

1. Hooks : permet d’incorporer des scripts a diverses étapes de la construction de l’ISOkali-config/common/hooks : hooks script <http://live.debian.net/manual/3.x/html/live-manual/index.en.html>
2. Include additional files or scripts in the build :

* Includes.binary : config fond d’écran delirs comme ça
* Includes.chroot : config root and usr

**THESE OPTIONS ARE TO APPLY IN THE APPROPRIATE SUB-DIRECTORY OF KALI-CONFIG**

* Lb config Noauto parameter is used to suppress a call to auto/config
* Clone a configuration via git

1. Customization package installation :

* With variant-name/package-lists/file.chroot you can only include packages that are already available in the official kali repository. If you have some of them not include you have to include the .deb files in a packages.chroot of the appropriate sub directory in the packages.chroot

Metapackages : <http://tools.kali.org/kali-metapackages>

* Kali-linux ( base system) : kali-desktop-common, apache2,mercurial,snmpd,mlocate,subversion,apt-transport-https,netcat,sudo,openssh-server,openvpn,tcpdump,git,snmp,whois,vim,php,php-mysql,……..
* Kali-desktop-live :
* Kali-linux-full : the default kali linux installation
* Kali-linux-all : meta package of all meta packages
* Kali-linux-sdr : software defined radio tools
* Kali-linux-gpu : GPU powered tools ( fro use of graphical card)
* Kali-linux-wireless : wireless assessment and analysis tools
* Kali-linuxWeb
* Kali-linux-Forensic
* Kali-linux-Voip : voice over ip
* Kali-linux-Pwtools : password cracking tools
* Kali-linux-Top10 : top10 popular tools : kali-linux,john,maltego,nmpa,aircrack-ng,maltego,sqlmap,burpsuite,malteo-teeth,wireshark,hydra,metasploit-framework,zaproxy
* Kali-linux-Rfid
* Kali-linux-netHunters
* Can include directly packages as files
* Package sources :
  + --distribution option : build the ‘option’ distribution
  + --mode : by default set to debian ;
* To configure additional repository : /config/archives/your-repository.chroot
  + Ex : config/archives/live.list.chroot : install package from the debian-live snapshot repository
* Choosing packages to install : 3 way

1. Using Packages lists
2. Using metapackages : populate a task metapackages maintaned by your distribution ex : echo task-gnome-desktop > config/package-lists/desktop.list.chroot
3. Local package lists : in config/package-lists : package with a .list suffix, and then an additional stage suffix .chroot or .binary to indicate which stage the list is for
4. Using conditionals inside package lists in config/\*
5. If both a live and an install list exist the packages in .list.chroot\_live list are removed
6. Custom kernel : has to be integrated within the debian package (.deb package) with a minimum requirements : initial ramdisk / union filesystem module (aufs) / include any other filesystem module required by your configuration( squashfs)

* Install modified or third party packages :
  + **In config/packages.chroot/ : simply copy it to the config/packages.chroot/** 
    - Must be named in the prescribed way ( easy with using dpkg-name)
  + Using a custom APT repository :
    - Apt(if a missing package is specified, package installation will fail) or aptitude ( missing package specified, package installation will succeed)

1. Customize contents :

* **Includes : allow to add or replacy arbitrary files by 2 mechanism :**
  + **Live/Chroot local includes : add or replace files to the chroot/live filesystem.** 
    - **Simply add them to your config/includes.chroot**

**Ex : to add /var/www/index.html :**

**mkdir -p config/includes.chroot/var/www  
$ cp /path/to/my/index.html config/includes.chroot/var/www**

* + **Binary local includes : for documentation or videos add or replace files in the binary image in : config/includes.binary/ this files will now appear in the root directory of the live medium**
* Hooks allow to execute arbitrary commands at different stages of the build and at boot time :
* Preseed : allow to configure packages when they are installed , files in the config/preseed/ directory suffixed with .cfg , followed by the stage .chroot,.binary and are installed using debconf-set-selections

1. **Make the build faster : put in cache the packages locally : install apt-cacher-ng and configure the locale variable http\_proxy**

* **Install apt-cacher-ng** 
  + **Start the service : /etc/init.d/apt-cacher-ng start**
  + **Create the locale variable http\_proxy : export http\_proxy=http://localhost :3142**
  + **Start the config by using the proxy : lb config –apt-http-proxy** [**http://127.0.0.13142/**](http://127.0.0.13142/)
  + **Launch the build**
* Utiliser squid comme proxy :
  + /etc/init.d/squid start
  + Run build.sh with –apt-http-proxy parameter :
    - Export http\_proxy=http://localhost :3128/
    - ./build.sh –distribution kali-rolling –verbose –variant custom -- \
    - --apt-http-proxy=${http\_proxy}

1. Examples :

* Tutorial image :
  + Mkdir tutorial1 ; cd tutorial1 , lb config
  + Lb build 2>&1 | tee build.log
* A web browser utility :
  + Mkdir tutorial2 ; cd tutorial2 ; lb config
  + Echo « task-lxde-desktop iceweasel » >> config/package-lists/my.list.chroot

Why ? lxde to provide a minimal desktop environment

* + Default configuration in config/includes.chroot/etc/iceweasel/profile/
  + Lb build 2>&1 | tee build.log
* A personalized image : keep a image in git and test multiples configurations using auto scripts
  + $ mkdir -p tutorial3/auto  
    $ cp /usr/share/doc/live-build/examples/auto/\* tutorial3/auto/  
    $ cd tutorial3
  + Edit auto/config to read as follows :

#!/bin/sh  
lb config noauto \  
     --architectures i386 \  
     --linux-flavours 686-pae \  
     "${@}"

* + Lb config
  + Populate local package list : $ echo "task-lxde-desktop iceweasel xchat" >> config/package-lists/my.list.chroot
  + Lb build
  + Git init && cp /usr/share/doc/live-build/examples/gitignore .gitignore
  + Git add . && git commit -m « Initial import »
* Second revision : clean up the first build , add vlc package , rebuild test commit
  + Lb clean
  + Add vlc packages : echo vlc >> config/package-lists/my.list.chroot
  + Lb build && git commit -a -m « Adding vlc media player »
* Make a minimal system :
  + Lb config –apt-recommends false
  + echo '! Packages Priority standard' > config/package-lists/standard.list.chroot
  + add packages needed : apt-cache depends live-config live-boot
    - if you discover you forgot some packages add them : $ echo "live-tools user-setup sudo eject" > config/package-lists/recommends.list.chroot
* A base image for 128MB USB key :
  + $ echo "live-tools user-setup sudo eject" > config/package-lists/recommends.list.chroot
  + To do more you can add : --debootstrap-options "--variant=minbase" and --firmware-chroot false and finally --memtest none
  + Add 2 other packages needed : echo "user-setup sudo" > config/package-lists/recommends.list.chroot
  + Lb build | tee build.log
* <https://www.cybrary.it/0p3n/create-kali-linux-iso/> :
  + Add custom tools : variant-XX/package.lists/kali.list.chroot : add wireshark , p0f ,hping3 and SET
  + Enable SSH service as boot : echo ‘update-rc.d -f ssh enable’ >> kali-config/common/hooks/01-start-ssh.chroot and make chmod 755 kali-config/common/hooks/01-start-ssh.chroot
  + Add unattended install option to not have to set options when we deploy the kali linux instance : kali-config/common/hooks/02-unattended-boot.binary + chmod 755 :

#!/bin/sh

cat >>binary/isolinux/install.cfg < label install

menu label ^Unattended Install

menu default

linux /install/vmlinuz

initrd /install/initrd.gz

append vga=788 — quiet file=/cdrom/install/preseed.cfg locale=en\_US keymap=us hostname=kali domain=local.lan

END

* + Add preseed file : wget https://www.kali.org/dojo/preseed.cfg  -O ./kali-config/common/includes.installer/preseed.cfg
  + Build : ./build.sh –variant XXX –distribution sana –verbose
  + If it’s in a VM : add the file to the apache2 server (don’t forget to start the service) in /var/www/html/dmz.iso

1. Stage of the build :
2. bootstrap stage :Populating the chroot directory with packages to make a barebones debian system
3. chroot stage : complete construction of chroot directory :
   * preseeds applied before any packages are instelled
   * package installed
   * hooks run later
4. binary stage : builds a bootable image using the content of the chroot directory
5. (if enabled) : source stage
6. Tools to install :
   * pwntools with pip :

apt-get update

apt-get install python2.7 python-pip python-dev git libssl-dev libffi-dev build-essential

pip install --upgrade pip

pip install --upgrade pwntools

* + Shellpop : list of reverse/bind shell without(or with) obsfuscation :

<https://github.com/0x00-0x00/shellpop>

root@kali# apt-get install python-argcomplete metasploit-framework -y

root@kali# pip install -r requirements.txt

shellpop –help

shellpop --list

* LIVE BUILD :
  + In Kali-config/common :
    - Bootloaders :
      * Grub-pc : configs for wallpapers and wallpapers
    - Hooks/live : script to enable persistence, forensice accessibility packages and configs per example cryptosetup
    - Includes.binary :
      * Isolinux : config background wallpaper
    - Includes.chroot :
      * Root : .bashrc
      * Usr : config user root and password and config ssh i don’t know why in user not in root
    - Includes.instalelr :
      * preseed.cfg : config replacing preseed.cfg embedded in intrd
      * usr/lib/live-installer.d/unhold/
    - packages-list : firmware ,architecture , linux-headers, packages standard
    - preseed :

