# Linux Penetration Testing Laptop (Xubuntu)

Boot the XUbuntu 7.04 Feisty Fawn cd.  It will boot into a live CD environment.

**NOTE:  The partitioning of the drive will fail if there is a windows install on the workstation(it is a BUG).  The window manager automounts the partition right when it is trying to partition it thus the error. You need to tell Thunar (window manager) not to automount the partitions.  Open Thunar->Edit->Preferences->Advanced Tab->Volume Management->Configure  Uncheck everything under the Storage tab.  Now start the install.**

Double-click the Install icon on the Desktop to begin the installation

Welcome [English]
Where are you? [NY EDT]
Keyboard layout [U.S. English]
Prepare disk space [Guided – use entire disk]
Migrate Documents and Settings [Nothing to do]
Who are you? [username:  linuxpen, computername:  linuxpen-laptop]
Ready to install

http://xubuntuguide.org is invaluable for getting Ubuntu configured post install.

**Updating and upgrading**

```
#apt-get update
#apt-get upgrade
```

**Adding compilers and kernel source code**

```
#sudo apt-get install build-essential
#sudo apt-get install linux-headers-`uname -r`
```

INSTALLING THE TOOLS

Create a source code directory to download all the code into

```
#cd
#mkdir source
#cd source
```

A lot of the source code downloaded will need some package libraries and development files installed before then can configure/compile

```
#apt-get install libssl-dev zlib1g-dev
```

### NMAP

Grab the latest install from www.insecure.org/nmap/

```
#wget http://download.insecure.org/nmap/dist/nmap-4.20.tar.bz2
#tar jxvf nmap-4.20.tar.bz2
#cd nmap-4.20
#./configure
#make && make install
```

### Host file parsing script

Various enumeration scans can leave you with numerous host list files.  A simple perl script will be created to allow combining, sorting, and listing of the unique ip addresses from your various lists.  Copy and past the following code to the file /usr/bin/unique.pl and make it executable #chmod 777 /usr/bin/unique.pl

```
#!/usr/bin/perl
foreach $argnum (0 .. $#ARGV){
        open (DAT, “$ARGV[$argnum]”) || die(“Could not openfile!”);
        while (<DAT>){
                chomp;
                push(@raw_data, $_);
        }
        close(DAT);
}
%seen = ();
@uniq = ();
foreach $item (@raw_data){
        unless($seen{$item}) {
                $seen{$item} = 1;
                push(@uniq, $item);
        }
}
@uniq = sort(@uniq);
foreach (@uniq){
        Print “$_\n”;
}
```

**SCAPY**
Download the latest version from http://www.secdev.org/projects/scapy/

```
#cd /usr/local/bin
#wget http://hg.secdev.org/scapy/raw-file/tip/scapy.py
#chmod 755 scapy.py
#cp scapy.py /usr/bin
#ln –s /usr/bin/scapy.py /usr/bin/scapy
#cp scapy.py /usr/X11R6/bin
#ln –s /usr/X11R6/bin/scapy.py /usr/X11R6/bin/scapy
#cp scapy.py /usr/bin/X11
#ln –s /usr/bin/X11/scapy.py /usr/bin/X11/scapy
#apt-get install python-pyx python-gnuplot python-crypto
#apt-get install imagemagick graphviz graphviz-dev libtools
#apt-get install tetex-base tetex-extra
```

NOTE:  accept all dependencies

**SCAPY Scripts:  tcpflags.py**

Copy and paste the code below into a file called tcpflags.py and save it in /usr/bin
Make it executable:  #chmod 777 /usr/bin/tcpflags.py
Create a symlink:  #ln –s /usr/bin/tcpflags.py /usr/bin/tcpflags

```
#!/usr/bin/env python
from sys import argv, stdout, stderr, exit
from scapy import *

if len(sys.argv) != 4:
        print "Usage: ./tcpflags.py <target_file> <dst_ports> <sourceport>"
        print "EXAMPLE: ./tcpflags.py /home/user/hosts.txt 21,22,23,80,443 31337"
        sys.exit(1)

target_file = sys.argv[1]
try:
        tf=open(target_file, 'r')
except:
        stderr.write('Error opening file ' + target_file + ' for inclusion.\n')
        exit(1)

dports = sys.argv[2]
dports = dports.split(',')
sport = int(sys.argv[3])

#print output header
print "DST_IP DST_PORT SENT_FLAG SENT_SEQ REC_ACK REC_FLAG"
for target in tf:
        target = target.strip()
        for dport in dports:
                dport = int(dport)
                flags=range(256)
                for fl in flags:
                        p1=IP(dst=target)/TCP(dport=dport,sport=sport,flags=fl,seq=fl)
                        ans,unans=sr(p1,timeout=0.5,verbose=0)
                        for snd,rcv in ans:
                                        print snd.dst,snd.dport,snd.sprintf("%TCP.flags%"),snd.seq,rcv.ack,rcv.sprintf("%TCP.flags%")
```

**Nessus**

Dependencies
```
#apt-get install libssl0.9.7
```

As of version 3.0 Nessus is now a closed sources product with pre-compiled binaries available for download. The Debian Binary successfully installs on Ubuntu. We will be using the BETA version as it has a command-line (nessuscmd) tool that allows for quick scanning with a handful of plugins.

Download the latest BETA version from www.nessus.org (currently 3.1.4). Registration will be required to download. This will also give you a key to download all Nessus plugins.

Download Nessus-3.1.4-debian3_i386.deb

Double-click the item (in window manager) to install or from the command line: #dpkg –i Nessus-3.1.4-debian3_i386.deb

Post install configuration

```
#/opt/nessus/sbin/nessus-add-first-user
        Login : linuxpen
        Authentication (pass/cert) [pass] :  pass
        Login password :
        Login password (again) :
```

```
        User rules ^D (control-d)
        Is that ok ? (y/n) [y] y
#/opt/nessus/sbin/nessus-mkcert
        CA certificate life time in days [1460]: (accept default)
        Server certificate life time in days [365]: (accept default)
        Your country (two letter code) [FR]: US
        Your state or province name [none]: GA
        Your location (e.g. town) [Paris]: Atlanta
        Your organization [Nessus Users United]: DOAA
        Press [ENTER] to exit
#/opt/nessus/bin/nessus-fetch --register XXXX-XXXX-XXXX-XXXX-XXXX
```

NOTE: check your email (the one you registered with) for the key code

Download and install the NessusClient from www.nessus.org.  There is an Ubuntu version of the client (7.04 Feisty Fawn)

```
#dpkg –I NessusClient-3.0.0.beta3-ubuntu704.i386.deb
```

Start nessusd as a background process

```
#/opt/nessus/sbin/nessusd –D
```

### Firewalk

Download the latest package from http://www.packetfactory.net/firewalk/
Download libnet (http://www.packetfactory.net/libnet/dist/libnet.tar.gz) and libdnet (http://libdnet.sourceforge.net)

```
#apt-get install libpcap-dev libpcap0.7 libpcap0.7-dev
```

Untar each of the tarballs

```
#tar zxvf *.tar.gz
#apt-get install flex m4 bison
#cd ../libdnet*
#./configure
#make && make install
#cd ../libnet
#./configure
#make && make install
```

```
#cd ../Firewalk
#./configure
```

Trying to compile at this point will give you an error. There is a bug in the code. If you attempt to compile you will note the line number given in firewalk.c where the error occurs (192 or 193).

```
#vim src/firewalk.c
```

Below the line /*empty*/ add **break;**
Make sure to include the semi-colon and save the file. You will be able to compile and install with no errors now.

```
#make && make install
```

**TRACEROUTE**

```
#apt-get install traceroute
```

**TCPTRACEROUTE**

```
#apt-get install tcptraceroute
```

**SCANRAND and PARATRACE**

```
#apt-get install paketto
```

**HPING**

```
#apt-get install hping2
```

**WIRE SNIFFING**

```
#apt-get install wireshark tcpdump
```

**UNICORNSCAN**

Unicornscan is difficult to configure to make it compile. However, there is an RPM package available. We will convert this package to an native DEB package so we can install it on Ubuntu. Download the binary RPM package from http://www.unicornscan.org

```
#wget http://www.unicornscan.org/releases/unicornscan-0.4.2-0.i386.rpm
#apt-get install alien
#alien –-to-dev unicornscan-0.4.2-0.i386.rpm
#dpkg –i  unicornscan-0.4.2-0.i386.deb
#ln –s /usr/lib/libpcap.so.0.8 /usr/lib/libpcap.so.0.8.3
```

**METAPLOIT**

Download the 3.0 framework

```
#apt-get install ruby openssl libopenssl-ruby
```

Accept all dependencies

**VMWARE SERVER**

Download the latest vmware-server from www.vmware.com. Register the product to receive a key. VMWARE does not support Ubuntu by default and a patch needs to be downloaded (http://ftp.cvut.cz/vmware/vmware-any-any-update113.tar.gz).

```
#apt-get install xinetd
#tar –xzf /Path/To/VMware-server-1.0.3-xxx.tar.gz
#cd vmware-server-distrib
#./vmware-install.pl
```

Choose the defaults until it asks:

**Before running VMware Server for the first time, you need to configure it by invoking the following command "/usr/in/vmware-config.pl". Do you want this program to invode the command for you now? [yes]**

Enter No and quit the install. To install the patch cd back to the start directory

```
#cd ..
#tar zxvf /Path/To/vmware-any-any-update113.tar.gz
#cd vmware-any-any-update113
#./runme.pl
```

It should prompt you to run vmware-config.pl. Choose Yes. The install should complete. VMware can be run from the console by: #vmware

Configure vmware and install Windows XP SP2. See Windows Penetration Testing Laptop Setup . . .

**TRUECRYPT Encryption Software**

Download the Ubuntu 7.04 x86 version from http://www.truecrypt.org/downloads.php

```
#tar zxvf truecrypt-4.3a-ubuntu-7.04-x86.tar.gz
#cd truecrypt-4.3.a
#dpkg –i truecrypt_4.3a-0_i386.deb
```

Create a volume to use for audit data.

```
#truecrypt –-size 1000M –-encryption AES –-hash SHA-1 –-filesystem FAT –c workpapers.tc
```

Follow the rest of the prompts for normal volumne, password, and randomness created via mouse movement.
Mount the volume.

```
#mkdir /mnt/workpapers
#truecrypt workpapers.tc /mnt/workpapers
```