Getting Started with a Pentest Distro

Cheat Sheet

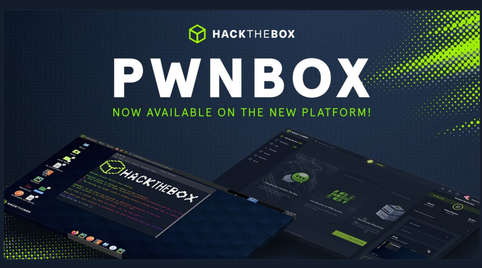
Anyone looking to start a technical path in information security must become comfortable with a wide range of technologies and operating systems. As penetration testers, we must understand how to set up, maintain, and secure both Linux and Windows attack machines. Depending on the client environment or scope of the assessment, we may be using a Linux or Windows VM on our machine, our base operating system, a cloud Linux box, a VM installed within the client's environment, or even perform testing directly from a client-owned workstation to simulate an insider threat (assume breach scenario).

Choosing a Distro

There are many Linux distributions (distros) for penetration testing. There are quite a few Debian-based pre-existing distros preloaded with many tools that we need to perform our assessments. Many of these tools are rarely required, and no distro contains every tool that we need to perform our assessments. As we learn and progress in our careers, we will gravitate to specific tools and have a list of "must-haves" to add to a new distro. As we progress, we may even prefer to fully customize our own pentesting VM from a Debian or Ubuntu base image, but building a fully custom VM is outside this module's scope.

The choice of a distro is individual, and, as mentioned, we can even choose to create and maintain our own from scratch. There are countless Linux distros out there that serve various purposes, some explicitly customized for penetration testing, others geared towards web application penetration testing, forensics, etc.

This section will cover setting up and working with Parrot OS. This distro is used for the Pwnbox that we will see throughout Academy, customized to practice and solve exercises throughout the various modules we will encounter.



It is important to note that each penetration test or security assessment must be performed from a freshly installed VM to avoid including security-relevant details from another client environment in our reports by accident or retaining client-sensitive data for significant lengths of time. For this reason, we must have the ability to quickly stand up a new pentest machine and have processes in place (automation, scripts, detailed procedures, etc.) for quickly setting up our distro(s) of choice for each assessment we perform.

Setting Up a Pentest Distro

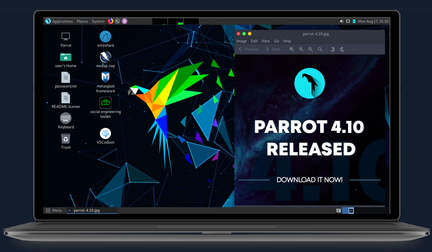
There are many ways to set up our local pentest distro. We can install it as our base operating system (though not recommended), configure our workstation to dual boot (time-consuming to switch back and forth between operating systems), or install using virtualization.

There are quite a few options available to us: Hyper-V on Windows, as virtual machines on bare metal hypervisors such as Proxmox or VMware ESXi or using free hypervisors such as VirtualBox, or VMware Workstation Player, which can be installed and used as hypervisors on Windows and Linux operating systems.
Another option is VMware Workstation, which requires a paid license but offers many more features than the free options.

A `hypervisor` is software that allows us to create and run virtual machines (VMs). It will enable us to use our host computer (desktop or laptop) to run multiple VMs by virtually sharing memory and processing resources.
VMs on a hypervisor run isolated from the primary operating system, which offers a layer of isolation and protection between our production network and vulnerable networks, such as Hack The Box, or when connecting to client environments from a VM (though VM breakout vulnerabilities do arise from time to time).

Depending on the amount of resources our host system has (i.e., RAM), we can usually run a few VMs at once. It is often helpful to stand up a VM during an assessment to test out an exploit or attempt to recreate a target application and stand-up machines in a lab environment to test out the latest tools, exploits, and techniques. Everyone working in a technical information security role should be comfortable working with one or more hypervisors and building virtual machines competently for both work and practice.

To be successful, we must continuously work to hone our craft. A great way is by setting up a home lab to attempt to reproduce vulnerabilities, set up vulnerable applications and services, see the effects of remediation recommendations, and have a safe place to practice new attack techniques/exploits. We can build our lab on an old laptop or desktop but preferably using a server to install a bare-metal hypervisor.



For our purposes, we will be using a modified version of Parrot Security (Pwnbox), available here to build a local virtual machine. We can choose two formats:

- `Optical disc image (ISO)`
- `Open Virtual Appliance (OVA)`

ISO

The `ISO` file is essentially just a CD-ROM that can be mounted within our hypervisor of choice to build the VM by installing the operating system ourselves. An `ISO` gives us more room for customization, e.g., keyboard layout, locale, desktop environment switch, custom partitioning, etc., and therefore a more granular approach when setting up our attack VM.

OVA

The `OVA` file is a pre-built virtual appliance that contains an `OVF` XML file that specifies the VM hardware settings and a `VMDK`, which is the virtual disk that the operating system is installed on. An `OVA` is pre-built and therefore can be rapidly deployed to get up and running quicker.

Once up and running, we can begin exploring the operating system, becoming familiar with the tools, and performing any desired customizations. The Parrot Linux team maintains a variety of helpful documentation:

- `What is Parrot?`
- `Installation`
- `Configuration`

For other questions, the Parrot team maintains a forum.

Practicing with Parrot

We will encounter Parrot Linux throughout Academy. The in-browser version, or Pwnbox, is available in any module sections which require interaction with a target host in our lab environment. Click the `Start Instance` button below and start becoming familiar with the Pwnbox. All interactive Module sections can be completed from our own VM after either spawning a Docker image or spawning a target host or multiple hosts and downloading a VPN key. Using the Pwnbox is not a requirement but is useful because all Academy work can be completed within our browser without requiring any virtualization software or additional resources to run a virtual machine.

Docker instances can be accessed without requiring a separate VPN connection. Specific hosts (i.e., Active Directory targets) require VPN access if not accessed from the Pwnbox. If this is the case, a button will appear to download a VPN key after spawning the target. We will begin working with target hosts later in this module.

Start Instance

1 / 1 spawns left

Waiting to start...

My Workstation

OFFLINE

Start Instance

1 / 1 spawns left