

Building a Hardware and Software Test Platform

This book is designed for those who need to better understand the importance of IT security. This chapter walks you through what you need to set up a hardware/software test platform. As a child, you may have loved to take things apart, TVs, radios, computers, and so on, in a quest to better understand how they worked. Your tools probably included soldering irons, screwdrivers—maybe even a hammer! That is similar to what you will be doing throughout this book. While you won't be using a hammer, you will be looking at protocols and applications to understand how they work. You will also examine some common tools that will make your analysis easier. The objective is to help you become a better network analyst, and improve and sharpen your IT security skills.

Because no two networks are the same, and because they change over time, it is impossible to come up with a one-size-fits-all list of hardware and software that will do the job for you. Networks serve the enterprises that own them, and enterprises must change over time. In addition, the scale of operation impacts security considerations. If you pursue a career as a security consultant, your goals (and inevitably your needs) will differ, depending on whether you work for a large multinational corporation (and even here, your goals and needs will depend on the type of industry) or a small office/home office (SOHO) operation or a small business. Clearly, a whole spectrum of possibilities exists here.

This chapter provides the first step in building your own network security lab. You will start to examine the types of hardware and gear that you can use

to build such a test environment, and then look at the operating systems and software you should consider loading on your new equipment.

Why Build a Lab?

A laboratory is as vital to a computer-security specialist as it is to a chemist or biologist. It is the studio in which you can control a large number of variables that come to bear upon the outcome of your experiments. And network security, especially, is a field in which the researcher must understand how a diverse range of technologies behave at many levels. For a moment, just consider the importance of the production network to most organizations. They must rely on an always-on functioning, which means that many tests and evaluations must be developed in a lab on a network that has been specifically designed for such experiments.

NOTE A laboratory is a controlled environment in which unexpected events are nonexistent or at least minimized. Having a lab provides a consequence-free setting in which damage that might result from experimentation is localized (and can, it is hoped, be easily corrected).

Consider something as basic as patch management. Very few organizations move directly from downloading a patch to installing it in the production environment. The first step is to test the patch. The most agreed-upon way to accomplish this is to install it on a test network or system. This allows problems to be researched and compatibility ensured. You might also want to consider a typical penetration test. It may be that the penetration-testing team has developed a new exploit or written a specific piece of code for this unique assignment. Will the team begin by deploying this code on the client's network? Hopefully not. The typical approach would be to deploy the code on a test network to verify that it will function as designed. The last thing the penetration test team needs is to be responsible for a major outage on the client's network. These types of events are not good for future business.

Building a lab requires you to become familiar with the basics of wiring, signal distribution, switching, and routing. You also need to understand how you might tap into a data stream to analyze or, potentially, attack the network. The mix of common network protocols must be understood; only by knowing what is normal on the network can you recognize and isolate strange behavior. Consider some of the other items that might motivate you to construct such a lab:

- Certification
- Job advancement
- Knowledge

- Experimentation
- Evaluation of new tools

To varying degrees, networking- and security-related certifications require knowledge of the hardware and software of modern networks. There is no better vehicle for learning about networking and security issues firsthand than to design and build your own network lab. This provides a place where you can add and remove devices at will and reconfigure hardware and software to your liking. You can observe the interaction between the systems and networking devices in detail.

Advancing in your field is almost never an accident. The IT industry is an area of constant change, and the best way to build a career path in the world of IT is to build your skill set. By mastering these technologies, you will be able to identify the knowledgeable people on the job or at a customer's site, and align yourself with them. You might even uncover some gifts that you did not previously realize you possessed, such as a love for hexadecimal—well, maybe.

Building a lab demonstrates your desire and ability to study and control networks. One key item that potential employers always consider is whether a candidate has the drive to get the job done. Building your own security lab can help demonstrate to employers that you are looking for more than just a job: You want a career. As you use the network resources in your lab, you will invariably add to your knowledge and understanding of the technologies that you employ. Learning is a natural consequence.

Experimentation is a practical necessity if you are to fully understand many of the tools and methods employed by security professionals and hackers alike. Just consider the fact that there are many manuals that explain how Windows Server 2012 works, or how a Check Point firewall works, but no manual can account for every single situation and what is 'unique' to any environment you encounter. Some combinations and interactions are simply unknown. By building your own lab, you will discover that when deployed in complex modern networks, many things do not work the way the documentation says they will. And many times, it does not suffice to simply understand *what* happens; you need to appreciate the timing and sequence of events. This requires the control that a laboratory environment provides.

Because IT is an industry of continual change, new software, new security tools, new hacking techniques, and new networking gizmos constantly appear. A network security lab provides you with a forum in which to try these things out. You certainly don't want to risk corrupting a computer that you depend on every day to do your job. And you don't want to negatively impact the work of others; doing so is a good way to quickly put the brakes on your budding career.

A laboratory thus provides a place where you can try new things. This is a setting in which you can gain a detailed understanding of how things are put together and how they normally interact. It is an environment in which you can likely predict the outcome of your experiments, and if an outcome is unexpected, you can then isolate the cause.

BUILDING YOUR OWN SECURITY LAB

A common question among students and those preparing for certification is, “How do I really prepare for the job or promotion I am seeking?” The answer is always the same: know the material, but also get all the hands-on experience you can. Many times they don’t have enough money in their IT budget, or they are a struggling student. That is totally understandable. Yet the fact remains that there is no way to pick up many of the needed skills by reading alone. And many tests cannot be conducted on a live Internet-connected network.

With a little work and effort, you can find the equipment required to practice necessary skills at a reasonable price—network professionals have been doing this for years. There are even sites such as certificationkits.com that are set up exclusively to provide students with a full set of networking gear needed to complete a Cisco Certified Network Associate (CCNA) or a Cisco Certified Network Professional (CCNP) certification.

Hardware Requirements

Before you can get started with any testing, you need to assemble some hardware. Your goal, as always, will be to do this as inexpensively as possible. Many things might be included in a network security laboratory. Some of these items are mandatory (for example, cables), and some things can be added according to your needs and as they become available or affordable. Although it is possible to contain everything within one computer, your requirements will vary from time to time based on the scenario that you are modeling.

Here are some of the things that will likely end up in your mix:

- Computers
- Networking tools
- Cables
- Network-attached storage (NAS)
- Hubs
- Switches
- Routers
- Removable disk storage
- Internet connection
- Cisco equipment
- Firewalls
- Wireless access points

- Keyboard, video, mouse (KVM) switches
- Surge suppressors and power strips

In your network lab, you will need a wide variety of cables, as this will allow you to configure your test network in many different ways. Specific configurations will be needed for different scenarios. You will also want to have some tools that come in handy for building and testing cables, so items such as wire strippers, crimp tools, and punch-down tools might find their way into your toolbox. Crossover and loopback adapters can prove handy, too.

Hubs, switches, and routers are the building blocks of network infrastructure. It is crucial to understand how the roles of these things differ. Not all switches have identical capabilities. Likewise, routers can vary considerably, so it is good to have a couple to choose from. Cisco products are so prevalent that it is a good idea to include some of their equipment in the mix; they will be found at almost every worksite.

An Internet connection is a necessity. You will need to research various topics and download software as you use the network in your lab. Or you might find yourself modeling the behavior of an Internet-based attacker. On the slim chance that you are borrowing WiFi from your neighbor's open access point, now is the time to make the upgrade to your own dedicated connection.

Having a firewall can prove very valuable, too. As a security professional, you are expected to have an appreciation for these devices and their capabilities. Your firewall could prove to be an important component in some of your experiments. On a daily basis, you can use your firewall to protect your primary (home or office) network from the unpleasant things that can occur on the network in your lab.

Don't forget the logistical details of constructing a network. You will need table space, shelving, power strips, and surge suppressors. If you have an old uninterruptible power supply (UPS) available, you might employ it, too. With several computers in close proximity, you will probably not want to have to deal with a bunch of monitors, keyboards, and mice; a KVM switching arrangement can save a lot of space and aggravation. Now you can turn your attention to the physical computing hardware that you will need.

NOTE Commercial-quality equipment is much more capable than the products targeted for the consumer or SOHO market. You will be better off with a real Cisco router, even if it is used and scratched up, than with a little Netgear home router.

Physical Hardware

When it comes to computer systems, there are three key items to consider: processor, memory, and disk space. Having a fast processor, a lot of memory, and a bunch of disk space is a big positive when selecting or building a computer.

Fast and *big* are relative terms whose meaning changes over time. But generally, a good place to start with a Windows PC would be an Intel Core i5 system with 32GB of RAM. Think of these as your minimum requirements. Generally, you can get away with a little less memory with Linux systems.

In terms of disk storage, an internal 1TB SATA hard drive would be considered a minimum requirement. While a solid-state hard drive is not mandatory, it will reduce boot-up times and it will reduce system response times. Removable disk storage, such as USB and NAS, can allow you to safely image your systems so that they can be restored with relative ease if they become corrupt during an experiment. NAS can be handy for holding copies of configuration files, downloaded software, and whatever else you may need while working on the network. It is great to have a central storage location that you can access from various computer systems.

So how do you start building your lab? First, consider many of the sources that exist for the equipment you need. Some of these sources include the following:

- Equipment you already have
- New equipment purchases
- Used equipment purchases

Each of these options is discussed in the following sections along with an overview of their advantages and disadvantages.

Equipment You Already Have

Either at home or at work, you are already likely to have some of the items that will prove useful in building your own security lab. These could range from something as trivial as a handful of Ethernet cables in your desk drawer to shelves full of spare or retired PCs, switches, and routers.

If you are doing this on the job, there are a couple of possible scenarios. Is the spare equipment under your control? If not, you will have to work things out with the appropriate supervisors and make sure that they approve your use of the equipment. Next, you want to take stock of what is available and make a list of the things that look like they could prove useful. Don't worry about the details at this point. Focus on the important items that were mentioned earlier in this chapter.

Finally, prioritize your list and pick out the things that you think will be most useful. Keep the list, as you will probably refer to it later. Remember to start with a small collection of obviously needed items, such as several PCs, laptops, a router, a hub or switch, an Internet connection, and a handful of cables. It will be easy to add things later, so try not to get carried away and include two of everything in your initial efforts.

New Equipment Purchases

Naturally, you have the option of buying new equipment. Sometimes this might be the easiest way to go, if you want to get the job done quickly. The only problem is that buying retail is probably the most expensive option. If you don't have much in the way of retired or spare equipment available, you might have to take this route. If you see your lab as a more or less permanent addition to the workplace, something that you plan to use on an ongoing basis for the foreseeable future, then maybe this is justified.

If you take this path, consider writing a proposal for the needed equipment. Determine the advantages that such a lab will bring to the department and to the company. Make sure to discuss these advantages in your proposal. Highlight the monetary savings that such an investment can return. On the positive side, this approach provides state-of-the-art equipment for the lab. You will also have all the manuals and software readily available. And you won't have to hunt around for missing parts. If you cannot get all the funds approved, you may decide that a few key components are best purchased new. Then the other odds and ends can be filled in on the cheap.

Of all the items that are recommended for inclusion in the lab, which one is best bought new? Many people would agree that PCs will most impact the usefulness of the lab. Older PCs tend to be somewhat slower and lacking in important resources, notably memory and storage capabilities. The prices of PCs have fallen considerably over the past few years. As an example, you can buy a decently equipped Dell "open source" desktop machine for around \$500. If you are going to put Linux on it anyway, you don't care that the machine does not come with an operating system. And if you intend to share one keyboard, display, and mouse with a KVM switch, again, who cares that the price does not include a display?

NOTE Watch the prices of memory and hard drives. Be careful with regard to memory prices if you decide to buy new computers. It is often cheaper to buy your own memory and install it in the machine yourself. And when it comes to hard drives, look for the breakpoint in the pricing where there seems to be an extraordinary price jump relative to the increase in drive size. That is the "sweet spot" in the market.

Used Equipment Purchases

If you are building your own security lab for home use, this may be the most viable option for obtaining some of the needed equipment. Although this route does require more work, you can save a substantial amount of money. It also spurs creativity, and that is a valuable skill in the networking and IT security field. Employ a bit of imagination. Who sells used computers, networking equipment, and pieces and parts? You will find no shortage of folks who sell

used items. Independent computer stores might have odds and ends that they would love to clear out of the way. You might encounter demonstration items or things that fall into the “open box” category. In retail, this is sometimes called B-stock. Some companies specialize in exactly this kind of thing. With a little web browsing, you are likely to discover several of them, such as www.liquidation.com and www.craigslist.com.

In addition, some flea market vendors specialize in used computer equipment. As an example, in Dallas, they hold a computer flea market twice a month. This is a paradise for computer nerds, who can likely find almost everything they need at a substantial discount. Check out www.sidewalksale.com if you’re going to be in the north Texas area. Other areas also set up such events; just ask around and check local resources. Who knows, you might find some useful items.

Computer companies often sell refurbished systems and components. Sometimes these items are returned by those challenged by a simple software or hardware problem (such as a missing software driver), or they have come back from a lease, or maybe there was a minor cosmetic defect or a trivial part was missing. Whatever the reason that motivates the seller, you can often find systems or significant components at prices that are well below retail. Some manufacturers outsource refurbished equipment that is returned. Often, the affected products are sold through various channels such as the Internet.

Although the risk is higher than with new equipment, the savings can be substantial. Just do your homework first. Check out the reviews for various items and determine whether others are reporting them as error prone or of high quality. Sites such as www.epinions.com and <http://reviews.cnet.com> report on specific products and hardware.

Online Auctions

eBay pioneered the online auction segment of the market back in the mid-1990s. Online auctions are a little different from the bidding process that you may be familiar with. Online auctions award the winning bid to the high bidder. This bid may have been placed three days before the auction’s closing, or three seconds before. Some individuals actually enjoy watching the last few seconds of the bidding process so that they can snipe the bid from another potential buyer just seconds before the auction ends. For the seller, a portion of the profits goes to the auction site in the form of seller fees. Buyers will want to look closely at any additional fees or charges that are placed on the final bid. Some individuals may even be running scam auctions in which they have no intention of ever sending you the goods purchased or may even misrepresent the goods as usable when they are in fact damaged. Here are some common tips for buyers:

- Bid low so that you don’t end up overpaying for the goods or services.
- Ask the seller questions if you want to know more about the item being sold.

- Monitor auctions close to their closing time to make sure that you don't miss a valuable item over a few dollars.

Online auction sites include www.liquidators.com, www.ubid.com, and www.ebay.com. eBay is the largest site and has proven to be an invaluable resource for buying and selling an endless number of things. They have a section dedicated to computers and networking, so if you are looking for a specific item, such as a particular brand and model of router, this is a super place to start your search. Even if you don't end up buying the item that you are interested in on eBay, you can get a good feel for the market price for whatever it is that you are curious about. It is very helpful to have a good sense of the cost of used items.

This book is not a forum for eBay do's and don'ts. Suffice it to say that you probably shouldn't buy anything off eBay that you are not prepared to write off as a loss. Although the vast majority of offerings are completely legitimate, horror stories do pop up from time to time. You must be the judge.

Be aware that while eBay transactions often avoid state sales taxes, these savings may well be offset by shipping and handling charges. Shipping may also take some time. Some sellers send items immediately after an auction closes, whereas others may wait days to ship. The time can vary considerably. This is not necessarily bad, just something to keep in mind if you have a project planned that is time-critical. All in all, eBay is a great resource. Just use common sense, and you will likely get a good result.

Thrift Stores

An often-overlooked option is thrift stores that handle used computer and network items. As an example, Goodwill has computer stores in Texas and California. The notion of recycling is often behind these operations. Businesses and individuals with old computers and related items donate them. The thrift organizations clean these components up, reformat the disk drives, strip some of the parts, and categorize them. If you're in a computer-centric area such as San Francisco, California, or Austin, Texas, these may be good places to find equipment to construct your lab. It is hard to say what kind of treasures you will find in these outlets. A thrift store might just have some equipment that is useful to you, such as the following:

- Hubs, commercial and consumer grade, single and dual speed
- Switches, likewise
- Routers, some of commercial quality
- Power bricks for many kinds of devices, including laptops
- SCSI adapters, cheap
- Ethernet network adapters (PCI and PCMCIA)
- CD and DVD drives, any kind you might need

- Monitors, many sizes, CRTs and LCDs
- Computer systems, both PC and Mac, with various operating systems
- Bare systems, comprising a case, power supply, Motherboard, CPU, memory, hard drive, and CD drive
- Old licensed software such as Windows Server 2003 or Windows XP that can be used to create target virtual machines

It is fair to assume that what is available varies from time to time with this sort of venue. Sometimes you will get lucky, and sometimes you will be disappointed. But the price is right.

Company Sales

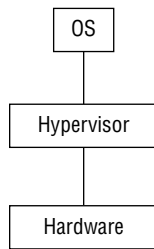
Many companies have employee sales from time to time. When this happens, employees have an opportunity to enjoy the first pick of equipment that is probably going to be donated, recycled, or discarded. It is often the case that the company is primarily interested in just getting rid of these items. They also see an additional benefit in making these things available to their employees. Making money is seldom a significant motivator. Large entities, government organizations, and schools do a lot of this type of activity. As an example, I attended one of these sales where Dell Latitude laptops were going for less than \$200 each. I was able to pick up 12 for use in a course kit I was building. The bottom line is, if you or one of your friends becomes aware of this kind of opportunity, you might want to take advantage of it.

Virtual Hardware

Modern computer systems have come a long way in how they process, store, and access information. One such advancement is in virtualization. While there are many types of virtualization, this section focuses on virtual systems. Virtual systems create an environment in which a guest operating system can function. This is made possible by the ability of the software to virtualize the computer hardware and needed services. Virtualized computing uses a virtual machine (VM), also called a virtual server. A VM is a virtualized computer that executes programs like a physical machine. VMware, VirtualBox, Virtual PC, Xen, and Hyper-V are a few examples of virtual machines.

A virtual server enables the user to run a second, third, fourth, or more operating systems on one physical computer. For example, a virtual machine will let you run another Windows OS, Linux x86, or any other OS that runs on an x86 processor and supports standard BIOS booting. Virtual machines are a huge trend and can be used for development and system administration and production, and to reduce the number of physical devices needed.

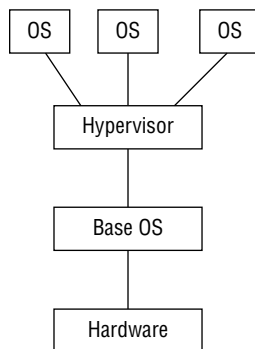
Virtual servers reside on a virtual emulation of the hardware layer. Using this virtualization technique, the guest has no knowledge of the host's operating system. Virtualized servers use hypervisors, which can be classified as either type 1 or type 2. Type 1 hypervisor systems do not need an underlying OS. This design of hypervisor runs directly on the hardware. An example of a type 1 hypervisor-based system is shown in Figure 1-1.



Native
(bare metal)

Figure 1-1: Type 1 hypervisors run directly on hardware.

A type 2 hypervisor runs on top of an underlying host operating system. The guest operating system then runs above the hypervisor. An example of a type 2 hypervisor is shown in Figure 1-2.



Hosted

Figure 1-2: Type 2 hypervisors run on an OS.

A type 2 hypervisor allows the physical system administrator to create guest operating systems that may be different from the base operating system. This technique uses a type 2 hypervisor to coordinate instructions to the CPU.

The hypervisor validates all the guest-issued CPU instructions and manages any executed code that requires additional privileges. VMware uses

the hypervisor, which is also known as a virtual machine monitor (VMM). The hypervisor is the foundation of this type of virtualization, as it accomplishes the following:

- Interfaces with hardware
- Intercepts system calls
- Operates with the operating system
- Offers hardware isolation
- Enables multi-environment protection

NOTE Two choices for virtualization include VirtualBox by Oracle and VMware.

This lab uses a type 2 hypervisor and Windows 7 for the base operating system, with several virtual systems loaded as guest operating systems.

VMware

Virtualization is the process of emulating hardware inside a virtual machine. This process of hardware emulation duplicates the physical architecture needed for the program or process to function. One of the first companies to develop a virtual product was VMware (www.vmware.com). They demonstrated this technology and patented it in the late 1990s. Before this, the development of hardware such as processors had not progressed enough to make this technology commercially viable for the average desktop-computer user. VMware would be a good choice to use in your lab because it enables you to easily test security tools, try out upgrades, and study for certification exams.

Probably the most important consideration is that more is always better. This means that more memory, more hard disk space, more processing power, and faster components always make for a better base system. You want to maintain a peak resource usage of no more than 60 percent to 80 percent. Greater usage will cause the systems to bottleneck and will also lead to performance problems. While VMware makes many different products, this section focuses on the following:

- VMware Player
- VMware Workstation

Table 1-1 lists some of the requirements and specifications of VMware products.

Table 1-1: Basic VMware Specifications

VIRTUAL DEVICE	PLAYER	PLAYER PRO	WORKSTATION
CD-ROM	Rewritable	Rewritable	Rewriteable
DVD-ROM	Readable	Readable	Readable

ISO mounting	Yes	Yes	Yes
Maximum memory	4GB	4GB	64GB
Processor	Same as host	Same as host	Same as host
IDE devices	4 max	4 max	4 max
NIC	10/100/1000	10/100/1000	10/100/1000
Video	SVGA	SVGA	SVGA
USB Support	3.0	3.0	3.0

As you can see in Table 1-1, VMware products include VMware Player, VMware Player Pro, and VMware Workstation. VMware Player runs on Microsoft Windows and Linux and can open and play any virtual machine created by another VMware product. One good thing about VMware Player is that it is free. The drawback is that it cannot create a virtual machine.

VMware Workstation is more advanced than VMware Player, and even supports an option known as snapshots, which means you can set a base point to which you can easily return. VMware Server is a much higher-end product; along with the added cost, it has the highest level of performance. For the lab setting you are building, VMware Workstation will work fine.

To install VMware Workstation, you need to either purchase a copy or download an evaluation copy. You need about 25MB of memory to download and install VMware Workstation. Just remember that this amount of memory is just to load the program. Each virtual system you install will require much more. On average, you will need a minimum of at least 8GB for each virtual OS you install. Memory is an important issue. Although the documentation might state that a minimum of 256MB of memory is needed, this typically won't be enough for anything more than a basic command-line install of Linux. Expect operating systems such as Windows to require much more. Insufficient memory will devastate performance on both the guest VM and host OS.

Here are the basic steps required to install VMware Workstation on the host OS:

1. Log on to your newly installed host OS as a user with Administrator privileges.
2. Find the newest VMware Workstation distribution at www.vmware.com/products/workstation/workstation-evaluation and then click the appropriate Download Now button, as shown in Figure 1-3. You need an e-mail address so that the key can be sent to you. If you do not want to purchase the program at this time, VMware will send you a key that is valid for 30 days.

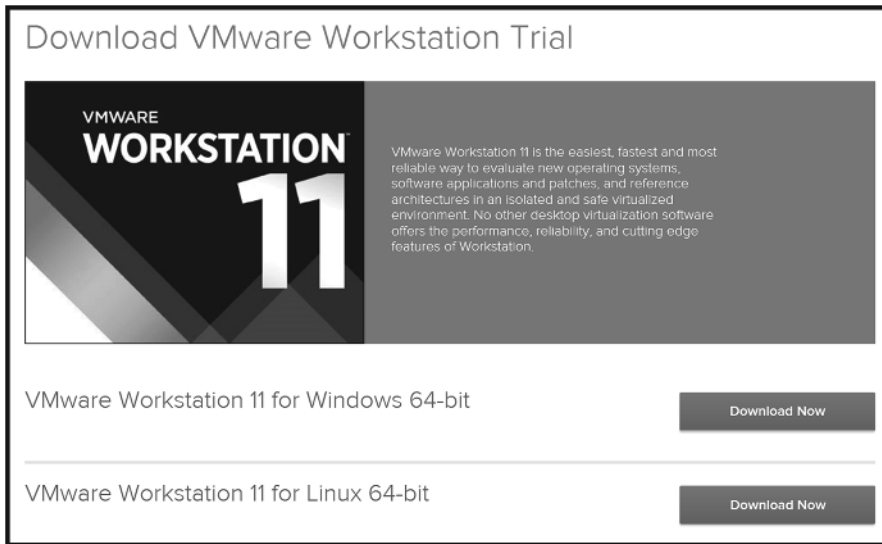


Figure 1-3: Install VMware Workstation.

3. Read the end-user license agreement, which explains the licensing terms. Click Yes to continue.
4. You are prompted to set the install location. The default is `C:\Program Files\VMware`. Keep this default unless you have a really good reason to change it.
5. Select any folder in which to install, and click Next. It takes a few minutes for the installer to create the necessary files on your system.
6. Because Windows systems use AutoRun for their CD/DVD players, the VMware installer asks whether you want to turn AutoRun off. You should say yes, because having it on can affect the functionality of the virtual machines.
7. If you have any previous versions of VMware Workstation, you are prompted to remove them. You are also prompted to create a VMware Workstation icon on your Windows desktop. Click Yes when prompted.
8. As with almost all Windows application installs, you are prompted to reboot your computer after the installation process is complete.
9. When the system reboots, VMware Workstation is installed. Opening the program displays a screen similar to that shown in Figure 1-4.

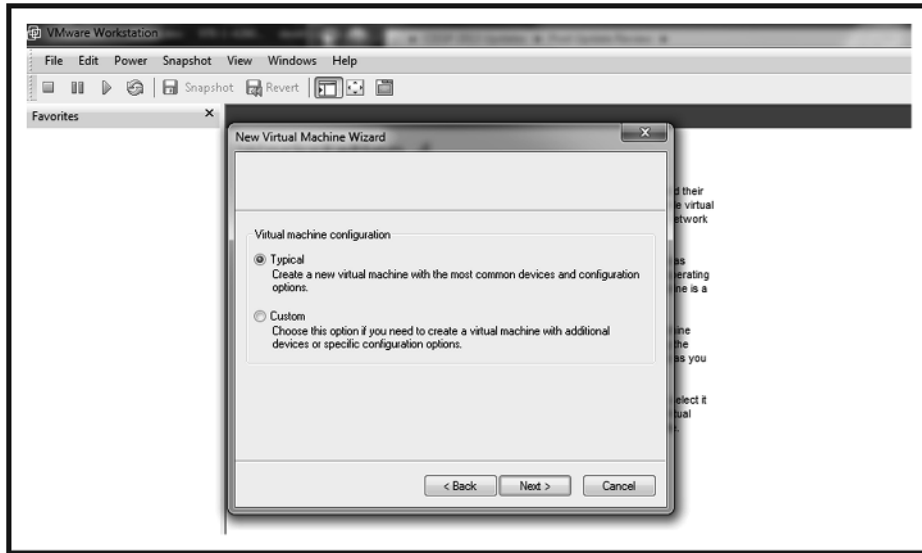


Figure 1-4: Choose the typical option to install the VMware Workstation.

NOTE Just because you have VMware Workstation installed doesn't mean that you are ready to start loading virtual operating systems. You must first enter a serial number. Remember that you can get a free, temporary 30-day evaluation license or buy a full license.

From this point forward, it is assumed that you have installed the files in the default location at `C:\Program Files\VMware\VMware Workstation`. In addition to a few shortcuts to Workstation, online help, and the uninstaller, you will find documentation in a compiled HTML help file for Internet Explorer or your browser located in the Workstation Programs folder: `VMware.chm`. If you look in the Programs directory, you will find a number of utility programs and auxiliary files such as `linux.iso`, `windows.iso`, and `freebsd.iso`. These ISOs contain the information used to install VMware Tools for Linux and Windows host systems. This will allow you to do things such as drag and drop files from the host OS to the virtual system. You don't need to transfer these files to actual CDs to use them; VMware will automatically attach them to the guest system when you perform a tools installation. You are prompted to do so after you install the virtual OS. The end-of-chapter exercises walk you through the installation of several different types of operating systems into VMware, such as Microsoft Windows and Linux.

VirtualBox

VirtualBox is the only professional virtualization solution that is freely available as open-source software under the terms of the GNU General Public License. It

is suitable for enterprise and home use, as well as a lab and testing environment. You will want to get started by downloading a copy from www.virtualbox.org/wiki/Downloads.

NOTE If you want to use a Mac, there are a few virtualization options, including VMware Fusion, Parallels Desktop, and VM VirtualBox.

Hacker Hardware

Most hacking gear is classified as software, but some hardware can be considered hacking gear, too, such as wireless cards, lock picks, key loggers, and phone taps.

One crucial piece of hacking gear is a WiFi adapter. Generally, the WiFi adapter on your PC will be insufficient for your lab. The key capability you need is to inject packets into the access point, and most built-in wireless adapters are incapable of packet injection. While there are many suitable options, one good choice is the Alfa AWUS036NH USB wireless adapter, which you can purchase for between \$30 and \$50. Some people like the AirPcap adapter but, it is more expensive, and is designed for Windows only. AirPcap adapters are used to capture 802.11 WLAN packets on Microsoft Windows computers. An AirPcap card can be used with tools such as Wireshark and Cain & Abel.

NOTE Aircrack-ng has a list of WiFi adapters that can work with their suite of tools. You can review the list at www.aircrack-ng.org/doku.php?id=compatibility_drivers&DokuWiki=69cd39e5bf14af0bfca2db56990ddb98.

Lock picks are another common category of physical hacking gear. Almost all hacking conferences feature some type of lock pick village. Contests are held to see who can pick a lock most quickly. Lock picks are used to open door locks, device locks, and padlocks. Most lock pickers don't learn lock-picking as a college course or through formal training; it is generally self-taught through practice. Lock-picking is really just the manipulation of a lock's components to open it without a key. The basic components used to pick locks are as follows:

- **Tension wrenches**—These are not much more than small, angled flathead screwdrivers. They come in various thicknesses and sizes.
- **Picks**—These are similar to a dentist's pick, and are small, angled, and pointed.

Together, these tools can be used to pick a lock. One of the easiest techniques to learn is *scrapping*. Scrapping occurs when tension is held on the lock with a tension wrench while the pins are scrapped quickly. A good site to learn more about locks is www.kickthefog.com/how_works.htm.

While this chapter may not go into an in-depth discussion on how lock-picking works, this is something that security professionals should know about. They

should also understand that it is important to check an organization's locks and make sure that they choose the right lock for the right job. Consider getting a lock-picking set to learn more about how lock-picking is actually performed. You may also want to get a set of bump keys, as shown in Figure 1-5.

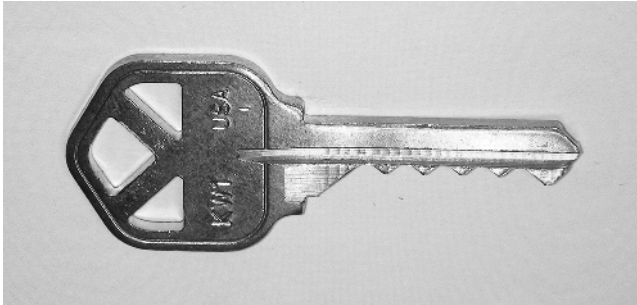


Figure 1-5: A bump key is a special key that has been cut to a number nine position and has a small amount of extra material shaved from the front and the shank of the key.

When slight pressure is applied and the key is bumped or tapped, this drives the pins upward and allows the attacker access. You will then be able to test your organization's physical defenses (with permission, of course).

Next on the list is keystroke loggers. A keystroke logger can be software or a hardware device that is used to monitor someone's computer activity. While an outsider might have some trouble installing one of these devices, an insider is in a prime position. Hardware keystroke loggers are usually installed while users are away from their desks, and they are completely undetectable except for their physical presence. Some loggers simply store the information and require you to retrieve them for analysis, while others have Bluetooth capability so that keystrokes can be wirelessly retrieved. To find such devices requires a physical inspection of the computer. And when was the last time you looked at the back of your computer?

Finally, this discussion isn't complete without some mention of phone-hacking tools. Actually, phone-hacking tools predate computer hacking. The 1960s and 1970s were the heyday of phone hacking. *Phreakers* (from "phone" and "freak") typically used phreak boxes (any device connected to a phone line) to perform their attacks. Some of the many types of phreak boxes (or color boxes) are listed here:

- **Blue box**—Enables you to make free long-distance calls
- **Red box**—Duplicates tones of coins being dropped into a pay phone
- **Tangerine box**—Used for eavesdropping without making a click when connected
- **Orange box**—Spoofs caller ID information on the called party's phone

Before you get too excited about making free phone calls, just remember that the use of these tools is illegal and that most of them do not work on modern telephone systems. The reason that much of this technology worked in the first place was because of in-band signaling. In-band signaling simply plays the control tones right into the voice channel onto the telephone wires. New telephone system networks use out-of-band (OOB) signaling, in which one channel is used for the voice conversation, and a separate channel is used for signaling. With OOB signaling, it is no longer possible to just play tones into the mouthpiece to signal equipment within the network.

CAP'N CRUNCH AND HIS BLUE BOX

John Draper was one of the first well-known phone hackers. His claim to fame was that he discovered how to use the toy whistle from a box of Cap'n Crunch. In the 1970s, long-distance phone service was still quite expensive—so much so that finding a way to make free calls was a pretty big deal. The exploit was actually possible because of the way the phone company handled signaling within the voice band of the call. Instead of relying on whistles to do this long-term, a small electronic box—named the *blue box*—was developed to handle just that task. This name is believed to be traced to the fact that the first one built was placed inside a small blue box. According to hacking legend, Steve Wozniak was so obsessed by the new technology that he called John Draper and asked if he could come visit him at his University of California, Berkeley, dorm and share his phone-hacking secrets.

Although the phreaking phenomenon slowed somewhat as technology changes enhanced telecommunication security, the culture never actually died, and phreaking lives on today in other forms. Today, a whole new generation has discovered things such as caller ID hacking. This phreaking technique gives an attacker the ability to make anyone's caller ID appear on the recipient's phone. Phone hacking also played a part in the News Corp UK phone hacking scandal of 2012.

Software Requirements

This section looks at software requirements, including operating systems and software. You may be asking yourself what the right operating systems are or how you will know which ones you need. If you are going to build your own network security lab, software will play a critical role. If you are building this lab with a tight budget, picking the right software will be even more critical, as there are certain pieces of software that you cannot live without.

One way to maximize your budget is by using virtual servers. This technology offers a great way to get more bang for your buck out of existing hardware. You will also look at some tools and applications that you might consider installing on your newly constructed operating systems. Finally, just remember the ultimate reason for using this type of test system: because you should never be running test software or experimenting on a production network. Unknown tools and software can cause many different results when combined with other software and processes. The worst case is when a critical system or service fails. You do not want to be the person who causes this to happen. For this reason alone, you should run a test lab on a nonproduction network.

Operating Systems

You cannot do a lot with the hardware you have until you load some software and operating systems. So, the following section discusses the types of operating systems to install and looks at the various options. Let's start by discussing the Microsoft family of operating systems.

Microsoft Windows

It almost goes without saying that any test network is going to need to run some version of a Windows system. Microsoft has helped redefine computing over the past 20 years. This history dates back to such classics as Windows 3.11 and Windows for Workgroups. This was one of the early top sellers for Microsoft and gave users a graphical interface along with the ability to network. In 1994, Microsoft released Windows NT 3.5, which was developed as a business-focused client/server operating system. Subsequent versions included Windows XP, Server 2003, Server 2008, Server 2012, Vista, Windows 7, Windows 8, and Windows 10.

The first question to consider is what version of Windows you should install. If you can find an old copy of 2003 server, this might be a good choice because there are a lot of exploits for this version. You should also consider Windows 7 because of its ubiquity in the corporate workplace. If you decide to install Windows 7, you first want to make sure that the hardware meets the minimum requirements:

- **Processor**—1 gigahertz (GHz) or faster 32-bit (x86) or 64-bit (x64) processor
- **Memory**—1 gigabyte (GB) RAM (32-bit) or 2GB RAM (64-bit)
- **Hard drive**—16GB (32-bit) or 20GB (64-bit) available hard disk space
- **Monitor**—VGA (800 × 600)
- **Disk drive**—CD-ROM or DVD
- **Other items**—Keyboard and mouse

Compare these requirements to those of Windows Server 2012, which are much greater:

- **Processor**—1.4 GHz 64-bit processor
- **Memory**—4GB RAM
- **Hard drive**—32GB available hard disk space
- **Graphics**—Super VGA (1024 × 768) or higher-resolution monitor
- **Disk drive**—DVD
- **Other items**—Internet access, keyboard, and mouse

As the preceding lists make very clear, it is much easier to meet the requirements for Windows 7 than for Windows Server 2012. For most of what is demonstrated in this book, Windows 7 will work fine. You may decide upon Windows 8, and it's certainly an option; just keep in mind that not everyone is a fan of the metro interface. Speaking of hardware, it is worth mentioning that Microsoft maintains a Hardware Compatibility List (HCL) at www.microsoft.com/whdc/hcl/default.aspx. This is a good site to check to make sure that your hardware is compatible before you begin installation. This is even more important if you have purchased used equipment.

If you're still unsure which software you should invest in, take a look at Table 1-2, which is a list of "must-haves" versus "nice-to-haves."

Table 1-2: Windows OS Priorities

OPERATING SYSTEM	COMMENTS
Windows XP	Acceptable for some testing of vulnerabilities but not a requirement
Windows 2003	Nice to have for demonstrating common vulnerabilities
Windows 7	Widely deployed; considered a must-have
Windows 8	Not widely deployed in the corporate environment
Windows Server 2012	Nice to have. Widely deployed in organizations using Windows servers

Linux

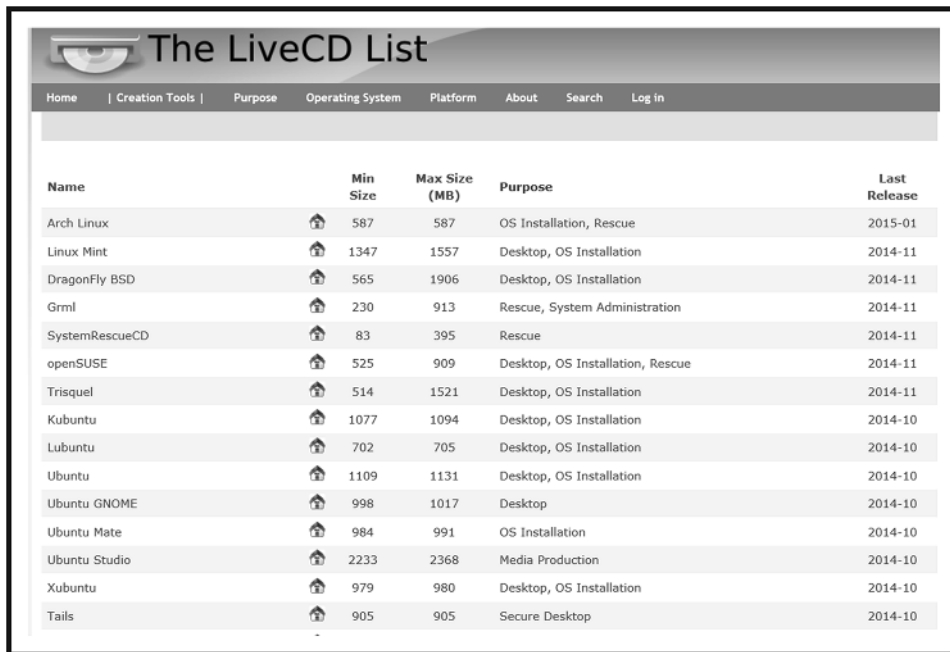
Linux is a Unix-like OS that can run from your Intel-based PC just like the Microsoft Windows OS. Linux was originally created by Linus Torvalds with help from programmers from around the world. If you're new to Linux, it is definitely an OS that you should get to know more about. The benefits of using Linux are that it is economical, is well designed, and offers good performance.

Linux distributions are easily available and can typically be downloaded for free. Linux comes in many flavors, including Red Hat, Debian, Mandrake, Ubuntu, and so on. Specialized versions have also been developed for specific purposes. Some of these include KNOPPIX, Fedora Security Spin, and Kali.

The best way to learn Linux is just by using it, which is why there is a copy of a Kali Linux downloadable version on the Wiley website. It is included as an *ISO image*. You can use the image to install Kali Linux onto a system or make a bootable DVD. If you are looking for other versions of Linux that have been customized for security work or to build your own security lab, you can review the list at www.livedcdlist.com/.

Linux is open source, which means that it can be freely distributed, and you have the right to modify the source code. It is also easy to develop your own programs on Linux. This is one of the reasons why you will see many security tools released on Linux well before they debut in the Windows world. This section of the chapter takes a closer look at installing Linux and reviews some of the basic features.

The easiest way to start is by using one of the bootable versions of Linux. As mentioned previously, www.livedcdlist.com has a good list that contains many of the most common distributions. You will find links to each specific version's website, as shown in Figure 1-6.



Name	Min Size	Max Size (MB)	Purpose	Last Release
Arch Linux	587	587	OS Installation, Rescue	2015-01
Linux Mint	1347	1557	Desktop, OS Installation	2014-11
DragonFly BSD	565	1906	Desktop, OS Installation	2014-11
Grml	230	913	Rescue, System Administration	2014-11
SystemRescueCD	83	395	Rescue	2014-11
openSUSE	525	909	Desktop, OS Installation, Rescue	2014-11
Trisquel	514	1521	Desktop, OS Installation	2014-11
Kubuntu	1077	1094	Desktop, OS Installation	2014-10
Lubuntu	702	705	Desktop, OS Installation	2014-10
Ubuntu	1109	1131	Desktop, OS Installation	2014-10
Ubuntu GNOME	998	1017	Desktop	2014-10
Ubuntu Mate	984	991	OS Installation	2014-10
Ubuntu Studio	2233	2368	Media Production	2014-10
Xubuntu	979	980	Desktop, OS Installation	2014-10
Tails	905	905	Secure Desktop	2014-10

Figure 1-6: Bootable security distributions of Linux

After you have selected any single distribution, you are taken to that version's download page.

The example in Figure 1-7 shows the Fedora Security Lab distribution.



Figure 1-7: Fedora Security Lab

When downloading an ISO, you may still need to perform an additional step or two to make the ISO useable. As an example, you may want to boot directly to the ISO from a DVD. The first thing you need to do is to convert the ISO into a bootable disk. This install uses a bootable CD-ROM; no installation to your hard drive is required.

To convert and use an ISO file from the Wiley website or one that has been downloaded from the Internet, you need the following:

- A CD/DVD writer
- A blank CD-ROM
- A burning program capable of burning an ISO onto a CD
- The capability to change your computer's BIOS to boot from the CD-ROM

A variety of Windows programs convert ISOs into bootable CD-ROMs, including Nero Ultra Edition, the ISO Recorder power toy, and Roxio Easy Media Creator Suite. If you have access to Mac OS X or a Unix or Unix-like workstation, these tools are already built into the base operating system. Here is a quick overview of the steps involved to complete the installation process.

1. If you are using Fedora Security Lab and you have only one CD/DVD drive, you need to copy Fedora Security Lab onto your hard drive before burning it to a blank CD. Otherwise, you can burn the image directly from the second CD-ROM drive.
2. Regardless of which tool you are using, open the application and select Burn Image to CD-ROM. When prompted for the image, select

`fedora-live-i686-21-5.iso`. If you are prompted to either Burn Disk at Once or Burn Track at Once, choose Burn Disk at Once.

3. When you have completed burning the CD, restart your computer, leaving the Kali CD in the CD-ROM drive. You might have to change the boot order in the BIOS by pressing F2 or the Del key during bootup.
4. After you have your computer set to the proper boot order, allow the computer to continue booting up.
5. Start Fedora Security Lab. Explore the interface; you will notice that there are many tools and applications. Some of these are discussed in later chapters.

Navigating in Linux

With Fedora Security Lab installed as a bootable disk, let's spend a few minutes discussing the basic structure of the OS and how it and other versions of Linux differ from Microsoft Windows. Some of the primary differences include the following:

- **Linux is case sensitive**—This is in contrast to Windows, which is not case sensitive. This means that `FAQ.txt` and `faq.txt` are two different files.
- **Linux directories and files have ownership permissions**—Linux uses the `Chmod` command to set permissions on files and directories. These can be restricted by user, group, and all others. Windows really has no equivalent to this command.
- **Regular Linux users cannot change system settings**—In the world of Linux, the all-powerful user is root. The root account has the ability to control critical settings. The closest thing that Windows has is the Administrator account.
- **Linux partitions are not based on FAT or NTFS**—Linux creates partitions using the `ext3` filesystem, whereas Windows uses FAT or NTFS partitions.
- **Linux path names contain forward slashes**—Unlike Windows, where a path might be `C:\Winnt\system32`, in Linux the path is `/var/log`.
- **Linux was developed for a multi-user environment**—This is much different from Windows because Windows evolved from DOS, which is a single-user operating system.
- **Linux does not use drive letters**—Whereas Windows uses drive letters, such as `A:`, `C:`, and `D:`, Linux contains everything within a single unified hierarchical structure.

The Linux filesystem is the structure in which all the information on the computer is stored. Files are stored within a hierarchy of directories; each directory

can contain other directories and files. Some of the more common directories found on a Linux system are as follows:

- `/`—Represents the root directory
- `/bin`—Contains common Linux user commands, such as `ls`, `sort`, `date`, and `chmod`
- `/dev`—Contains files representing access points to devices on your systems. These can include floppy disks, hard disks, and CD-ROMs.
- `/etc`—Contains administrative configuration files, the `passwd` file, and the `shadow` file
- `/home`—Contains the user's home directories
- `/mnt`—Provides a location for mounting devices such as CD-ROMs and floppy disks
- `/sbin`—Contains administrative commands and daemon processes
- `/usr`—Contains user documentation, graphical files, libraries, and a variety of other user and administrative commands and files

Directories and files on a Linux system are set up so that access can be controlled. When you log in to the system, you are identified by a user account. In addition to your user account, you may belong to a group or groups. Therefore, files can have permissions set for a user, a group, or others. For example, Red Hat Linux supports three default groups: super users, system users, and normal users. Access for each of these groups has three options:

- Read
- Write
- Execute

To see the current permissions, owner, and group for a file or directory, type the `ls -l` command. This displays the contents of the directory you are in with the privileges for the user, group, and all others. For example, the list of a file called `mikesfile` and the directory `mikesdir` would look like the following:

```
drwxr-xr-x  2 mikeg  users      32162 Aug  20 14:31 mikesdir
-rw-r--r--  1 mikeg  users       3106 Aug 16 15:21 mikesfile
```

The permissions are listed in the first column. The first letter indicates whether the item is a directory or a file. If the first letter is `d`, the item is a directory, as in the first item listed above, `mikesdir`. For the file `mikesfile`, the first character is a dash (`-`). The next nine characters for the `mikesdir` folder denote access and take the following form: `rwX|rwX|rwX`. The first

three characters list the access rights of the user, so for the `mikesdir` folder, the user has read, write, and execute privileges. The next three bits denote the group rights; therefore, the group has read and execute privileges for the `mikesdir` folder. Finally, the last three bits specify the access all others have to the `mikesdir` folder. In this case, they have read and execute privileges. The third column, `mikeg`, specifies the owner of the file or directory, and the fourth column, `users`, is the name of the group for the file or directory. The only one who can modify or delete any file in this directory is the owner, `mikeg`.

The `chmod` command is used by a file owner or administrator to change the definition of access permissions to a file or set of files. The `chmod` command can be used in symbolic and absolute modes. Symbolic mode deals with symbols such as `rwx`, whereas absolute mode deals with octal values. For each of the three sets of permissions on a file—read, write, and execute—read is assigned the number 4, write is assigned the number 2, and execute is assigned the number 1. To make permissions wide open for you, the group, and all users, the command would be as follows:

```
chmod 777 demofile
```

(This value is arrived at by adding 4, 2, and 1 together. Remember that 4 is for read, 2 is for write, and 1 is for execute.)

Linux Basics

The objective of this section is to review some Linux basics. Although a lot of work can be done from the Linux GUI, you will still have to operate from the Terminal window or shell. The Terminal window is similar to the command prompt in Windows. If you log in as root and open a Terminal window, you should see something similar to this: `[root@slax ~]#`. The `#` sign is most important here because it denotes that you are root. Root is god in the world of Linux. You want to make sure that you properly execute commands while working as root. Unlike Windows, Linux might not offer you several prompts or warnings before it executes a critical command.

It is important that you know some basic Linux commands and their functions. There are many, and so for the sake of brevity, Table 1-3 lists just a few basic commands. If all this talk of Linux commands has left you wanting more, you might want to spend a few minutes reviewing a more complete list of commands at the following sites:

- www.mediacollege.com/linux/command/linux-command.html
- www.laynetworks.com/linux.htm

Table 1-3: Basic Linux Commands

COMMAND	DESCRIPTION
/	Represents the Root directory
cat	Lists the contents of a file
cd	Changes the directory
chmod	Changes file and folder rights and ownership
cp	Runs the copy command
history	Shows the history of up to 500 commands
ifconfig	Similar to <code>ipconfig</code> in Windows provides network configuration
kill	Kills a running process by specifying the PID
ls	Lists the contents of a folder
man	Opens manual pages
mv	Moves files and directories
passwd	Changes your password
ps	Runs the process status command
pwd	Prints the working directory path
rm	Removes a file
rm -r	Removes a directory and all its contents
Ctrl+P	Pauses a program
Ctrl+B	Puts the current program into the background
Ctrl+Z	Puts the current program to sleep

Linux requires that user accounts have a password, but by default it will not prevent you from leaving a password set as blank. After installing BackTrack and while booting up, note that the default username and password is listed as *root* and *toor*. Linux encrypts the password for storage in the `/etc` folder. Most versions of Linux, including Kali, use MD5 by default. If you choose not to use MD5, you can choose DES, although it limits passwords to eight alphanumeric characters. Linux also includes the `/etc/shadow` file for additional password security. Moving the passwords to the `shadow` file makes it less likely that the encrypted password can be decrypted, because only the root user has access to the `shadow` file. If you are logged in as root and want to see the shadow passwords on your computer, execute the following command:

```
ls /etc/shadow
```

The format of the shadow file is

```
Account_name:Password>Last:Min:Max:Warn:Expire:Disable:Reserved
```

Linux systems also use *salts*. Salts are used to add a layer of randomness to the passwords. Because MD5 is a hashing algorithm, this means that if you use *topsecret* for your password and another user uses *topsecret* for their password, the encrypted values will look the same. A salt can be one of 4,096 values and helps further scramble the hashed password. Under Linux, the MD5 password is 32 characters long and begins with \$1\$. The characters between the first and second \$ represent the salt. Passwords created in this way are considered to be one-way. That is, there is no easy way to reverse the process. Figure 1-8 demonstrates how Linux creates this value.

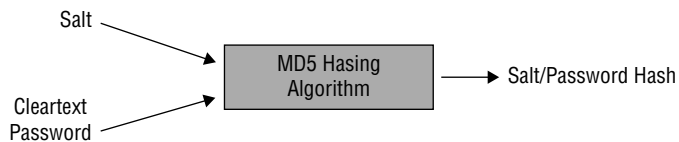


Figure 1-8: Linux password creation

SHADOWS VERSUS SALTS

The world of computing used to be a much more trusting place. At one time in the not-too-distant past, Linux passwords were kept in the `passwd` file. The `passwd` file is world-readable, which basically means that anyone can access or read this file. This means not only the people or processes you would like to read it, but also the bad guys. That is why the `shadow` file was created.

The `shadow` file is readable only by root. This helps keep the prying eyes of unauthorized users from taking a peek at encrypted passwords when they shouldn't be looking at them. Now, even if they do get a look at the passwords, the passwords are not formatted as unencrypted text; instead, they are kept in a hashed format. Hashes are considered one-way functions, as they are easy to compute in one direction yet very hard to compute in the other. The problem is that two identical words will create the same hash. That is why salts are needed, as they provide that second layer of randomness. A complete hash is made up of `1_SALT_$_HASH_`. The `$1$` refers to the algorithm being used—in this case, the MD5 algorithm. Salt lengths can vary; a common implementation is to use two random characters, which are stored as the first two characters of the encrypted password. For example, if `1yAkjfqifnips` is the encrypted value, then `1y` is the salt. That value is not only needed for the user to log on, but also for the attacker trying to crack the account.

Because the hashing process is one-way, there is no known way of directly retrieving the original password from the encrypted version. However, the attacker can extract the salt and use this two-character value to encrypt with a dictionary of words, and then compare those to the existing encrypted values. If the password happens to be a word in the dictionary, a match will be found and the password revealed.

Now that you are familiar with some Linux basics, let's look at the Mac OS X operating system.

Mac OS X

The Macintosh has always been considered innovative, ever since its introduction in 1984, but by the late 1990s it was due for an update. This update occurred by means of Mac OS X. The OS that had been developed by NeXT Software became the basis for OS X. OS X is a Unix/FreeBSD-based operating system designed to meet current and future computing needs. At the time of this book's publication, OS X is currently at version 10.10 Yosemite. With the release of 10.4.4, the operating system changed from supporting only PowerPC-based Macs to include Intel-based computers. Before you get too excited about running Mac OS X on your own Intel computer, Apple has stated that Mac OS X will not run on Intel-based personal computers aside from their own. As a result, OS X would require additional hardware. You will have to weigh the benefits and costs of investing in this technology.

When considering adding the Mac OS, take a look at the corporate environment in which you work. Some industries use Macs more than others. Schools, advertising agencies, and other industries that must perform graphics, video, and audio editing typically favor Macs. Some security professionals prefer Macs to PCs, and a growing number of end users are buying Macs, which somewhat parallels the growing popularity of Android.

ALPHA AND BETA SOFTWARE

The term *beta* is thought to have originated at IBM during the 1960s. Alpha tests are the first round of tests performed by the programmers and quality engineers to see how applications will function. Beta testing comes next. Beta testing is widely used throughout the software industry. This second round of product development has evolved to include testing that is performed internally and externally by prospective users.

While the software is potentially unstable, it is much more user-friendly than in its alpha stage, and gives the programmers, quality engineers, and users a good look at how the end product will act and perform. After collecting feedback from these initial users, the software is refined with another round of improvements before it is released in its final form.

Software and Applications

Installing an OS is only half the battle. After an OS has been installed, you need some client-side security tools to get any real work or exploration done. Security tools have been around for quite some time. Dan Farmer and Wietse Venema helped start the genre of security software in 1995 when they created one of the first vulnerability-assessment programs called Security Administrator Tool for Analyzing Networks (SATAN). This program set the

standard for many tools to follow; it made it possible to scan for vulnerable computers through the Internet and provided a variety of functions in one package. Although SATAN was a great tool for security administrators, it was also useful to hackers. That's the nature of tools; they can be used with good or bad intentions.

SATAN'S DAYS WERE NUMBERED

In 1995, few network-vulnerability tools existed. That is one reason why SATAN made waves in the world of network security. The debate at the time centered on the real purpose of the tool. Was it designed for security administrators to verify security settings, or was it for attackers to use to scan for vulnerable systems that could be easily hacked? This debate was further fueled by the fact that in 1996 Dan Farmer performed a survey in which he scanned 2,200 Internet hosts with SATAN and found that more than half were vulnerable to attack. Not only were these systems scanned without the permission of the owners, but they were also not mom-and-pop sites. Mr. Farmer chose to scan high-profile sites such as banks and major institutions.

There is also the issue of the name of the program. To address those concerns, the install package actually contained a program named *repent*. This program would actually change all instances of the name "SATAN" to "SANTA."

SATAN was designed to run from a web browser. This made the tool easy to use and formatted the results in a summary fashion. While SATAN is considered outdated by today's standards, its contribution is that it spawned a segment of security software that did not previously exist. SATAN lives on today through such tools as SARA, SAINT, and Nessus.

Today, an untold number of client-side security tools can be used to scan for vulnerabilities, probe for holes, and assess security. Some of these are legitimate security tools, and others have been written by hackers or those without the best of intentions. As a security professional, you probably want to keep a variety of these tools handy. Just make sure that you have authorization before using them on a network.

Learning Applications

The final section of this chapter looks at some of the learning applications and hacking software that you can run in a lab environment to help you analyze common security problems and misconfigurations. The concept behind these learning applications is that these tools can help build your security skills. One good place to start is <https://www.vulnhub.com/>. This great website, shown in Figure 1-9, provides downloads, applications, and challenges that allow anyone to gain practical hands-on experience in application and network security. Creators of home labs will need to investigate limitations before investigating "hack me" sites. The AUPs of their subscription services may disallow any sort of hacking activities from subscribers.

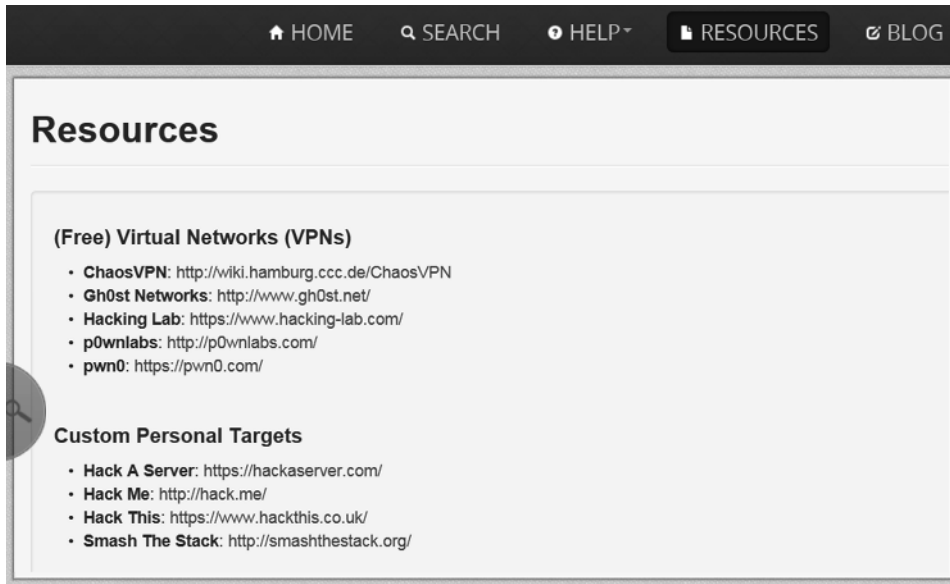


Figure 1-9: The Vulnhub website is useful to the security professional.

Next on the list is www.wechall.net/. This site maintains links to many different challenge sites. These sites focus on a lot of different types of challenges, such as hacking, cryptography, and steganography. If that's not enough to keep you busy for a while, here are two other sites worth investigating:

- **Hacme Bank**—A web-based bank that you can actually hack without worrying that you will go to jail
- **Damn Vulnerable Linux**—A Linux distribution that is packed with vulnerable applications

Hacme Bank works in much the same fashion as previously listed tools. It is available from Foundstone at www.foundstone.com/us/resources-free-tools.asp. This application also installs a simulated bank that is designed to teach you how to create secure software. Hacme Bank has an assortment of common vulnerabilities built in, such as SQL injection and cross-site scripting. This tool is actually used in Foundstone security classes.

Damn Vulnerable Linux may not be one of the newest hacking distributions, but it remains popular. It is available for download at <http://distrowatch.com/table.php?distribution=dvl>. Damn Vulnerable Linux has been loaded with broken, buggy, outdated, and exploitable software. Its primary goal is to design a Linux system that is as vulnerable as possible to allow individuals like you who are building a lab to explore code injection, buffer overflows, shell code development, web exploitation, and SQL injection.

Hacking Software

While the title of this section may have gotten your attention, it actually refers to a range of software and applications that are widely used by hackers and security professionals alike. In effect, by building a network lab, you are creating an environment in which you can (and must) ethically hack. And while on this topic, it should also be made clear that you should never run any tools or exploits on an outside or external network without the network owner's permission. The objective is to keep it legal while you increase your knowledge.

Many pieces of software can be used for good or malicious purposes. For example, consider port scanners. While attackers use them to scan open ports that can be used for potential attacks, security professionals use port scanners to verify that ports truly are closed and that firewall rule sets are working. Therefore, if you were going to make a short list of dual-use software, you might include the items in the following list.

The best place to start gathering tools is <http://sectools.org>. This site, run by Insecure.Org, lists the top security tools, and has done so since 2000. Check out the site for a complete listing, but in the meantime here are the top ten:

- **Wireshark**—Packet sniffer
- **Metasploit**—Exploit framework
- **Nessus**—Vulnerability assessment tool
- **Aircrack**—Wireless exploitation tool
- **Cain & Abel**—Diverse Windows exploitation tool
- **Netcat**—Command-line back-end tunneling tool
- **tcpdump**—Packet sniffer
- **John the Ripper**—Password-recovery tool
- **Kismet**—Wireless hacking tool
- **Burp Suite**—Web proxy and web application tester

There are also several tools that deserve honorable mention:

- OWASP Web Proxy
- Capsa Network Analyzer
- Nmap
- BeEF browser exploit framework
- IDA Pro
- OWASP Xenotix Exploit Framework
- FOCA Network Intelligence tool

A lot of other hacking tools are available, yet many, such as virus generators or remote access Trojans (RATs), have little or no practical purpose other than to spread malware and cause problems. This book won't spend much time examining these types of tools, but just keep in mind they do exist.

Summary

Building your own security lab to serve as a laboratory environment for network security experimentation is not difficult to do, and it need not be particularly expensive. By applying some effort and taking a little time, you can cut your costs and still build a good test bed. By using some of the things that are likely already available to you and adding a few additional components, you can build a network in a couple of days. The benefits are many. First, this provides a setting in which you can work with hacking tools without impacting other network users. If damage occurs, and you built the network intelligently, used virtual images, and backed-up everything, it will be relatively easy to restore systems to their previous state.

One key piece of this project is determining which operating systems to install. Just because of their dominance in the marketplace, you need to install Windows and Linux operating systems. Windows is the most popular desktop OS and is used extensively around the world. Understanding its vulnerabilities and how it is secured is an important component of building your own security lab. Linux is well positioned as a backend server for many major firms around the world. Linux is also an important platform for security tool development. Much of this is based on the open source nature of the OS. Open source means that you can search for a fix and even solicit the user community for help. Much like distributed computing, the result is that you have thousands of eyes and minds working on problems and glitches.

Another important topic in this chapter concerned how to do more with less. This means a way to have more computer operating systems running with fewer physical computers. This is what virtualization allows a user to do: to use one host system to support many virtual operating systems. Several options were discussed, but in the end, selecting one to use is very much a personal choice. The book itself is focused on VMware because the VMware player is free and because VMware has a lot of industry support. It has proven itself to be a robust virtualization product. However, if you prefer to go the open source route, you might want to look at alternatives such as VirtualBox.

Finally, the chapter looked at some learning applications. These included options such as Damn Vulnerable Linux. These distributions enable you to set up a complex environment, such as an online bank, and look at the processes and interactions between the client and server. The idea is to learn what works well and what is

potentially vulnerable. The intention of this chapter was to help you set up the software and hardware platform you will be using for the rest of this book and, as you continue to use your lab, to learn more about networks and security controls.

Key Terms

- **Chmod**—A Linux command that is used to change the mode of a file.
- **etc/shadow file**—One possible location of the Linux password file (`/etc/shadow`), which is only accessible by root.
- **Firewall**—A hardware or software security system that is used to manage and control both network connectivity and network services. Firewalls act as chokepoints for traffic entering and leaving the network, and prevent unrestricted access. Firewalls can be stateful or stateless.
- **Hub**—A device that connects the cables from computers and other devices such as network-attached storage in an Ethernet LAN.
- **ISO image**—A CD or DVD disk image that can be stored as a single file yet represents the complete structure of an optical disk.
- **Lock-picking**—The art of opening locks without the keys.
- **Mandatory access control**—A means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (such as clearance) of subjects to access information of such sensitivity.
- **MD5sum**—A cryptographic algorithm that is used to verify data integrity through the creation of a 128-bit message digest.
- **Network-attached storage**—A device that is accessible directly on the LAN and is designed for handling files and data storage.
- **Phreaking**—A term used for individuals who crack telecommunication security, most often phone or voice communication networks.
- **Router**—A device that determines the next network point to which a data packet should be forwarded en route to its destination. Routers create or maintain a table of the available routes and use this information to determine the best route for a given data packet. Routing occurs at Layer 3 (network layer) of the OSI seven-layer model.
- **Salt**—A random string of data used to modify a password hash to provide randomness to stored passwords.
- **Switch**—A device that links several separate LANs and provides packet filtering between them. A LAN switch is a device with multiple ports, each of which can support an entire Ethernet or Token Ring LAN.

- **Virtualization**—Creation of a software implementation of a hardware device. Virtualization enables users to run multiple operating systems on the same physical computer in isolation from each other.
- **WiFi detectors**—Devices designed to detect wireless signals.
- **Wireless access point**—A device used to bridge a wired and wireless network. Wireless access points act as a central node for users of wireless devices to connect to a wired network.

Exercises

This section presents several hands-on exercises to help reinforce your knowledge and understanding of this chapter. The tools and utilities used in these exercises were chosen because they are easily obtainable. The goal is to provide you with *real* hands-on experience. The most important exercise to complete at the end of this chapter is to build your network. Because equipment varies and many different designs are possible, it is hoped that you take this time to construct a hardware base to use for subsequent chapters.

Equipment Checklist

For this first exercise, fill in the following checklist of items that need to be completed to get your lab ready for software installation.

ITEM	DESCRIPTION	DATE COMPLETED
1	Select a location for the lab.	
2	Specify the floor space needed and any added environmental requirements such as air conditioning.	
3	Specify the external network connections.	
4	Determine the computer and server hardware requirements.	
5	Determine required operating systems.	
6	Determine required application software.	
7	Determine any utilities or other software required.	
8	Determine needed tools and test equipment.	
9	Determine network cabling and network equipment required.	
10	Acquire the workspace needed for the lab.	
11	Have any required power, phone, network cabling, and external network connections installed.	

12	Obtain the network infrastructure hardware, computer hardware, software, tools, and test equipment.
13	Set up the network.
14	Set up the computers and servers.

NOTE For this book, a Toshiba Satellite L70-BBT2N22 Laptop with Windows 7 Professional is used. It is equipped with 16GB of DDR3I RAM, with a 1TB solid-state drive. This will be used as a test platform. A Buffalo LinkStation 2TB High Performance NAS will also be used for additional storage.

Installing VMware Workstation

1. Download VMware Workstation.
2. Double-click the application to start the installation.
3. Once installation is complete, enter the serial number key when prompted.
4. Explore some of the options of VMware Workstation.

Exploring Linux Operating System Options

One of the great things about virtualization is the ability to set up virtual machines. Check out www.vmware.com/ and explore some of the ready-to-use images that are available to download. There are also several Linux ISOs that I recommend you install with VMware Workstation or VirtualBox.

OS	VERSION/DESCRIPTION	SIZE
Kali	https://www.kali.org/downloads/	2.9GB
Fedora Security Spin	http://spins.fedoraproject.org/security/	890MB
Damn Vulnerable Linux	http://sourceforge.net/projects/virtualhacking/files/os/dvl/damnvulnerablelinux_1.0.iso/download	149.6MB

NOTE These three Linux distributions will give you a good set of Linux-based VMs for testing. If you only have enough storage space for one, I recommend that you install Kali.

Using VMware to Build a Windows Image

This first exercise steps you through a Windows 2003 installation. Windows 2003 was chosen for this example because it's old, has numerous vulnerabilities, and

will work well to demonstrate exploits in later chapters. The licensed, sealed copy of Windows 2003 Server used in this example was purchased from eBay for only \$12.00.

1. Open VMware.
2. Choose New Virtual Machine and let the wizard step you through the setup.
3. Select the default setting until you get to Select a Guest Operating System. Choose Microsoft Windows and Windows 2003 Server.

NOTE If you don't have a copy of Windows 2003, you can download a trial of some Windows products at <https://www.modern.ie/en-us/virtualization-tools#downloads>.

4. Continue to accept the defaults. You are prompted for bridged network and default disk size. The default setting should be good for both of these. When the wizard is finished, you are presented with the Windows 2003 Server tab. You now want to insert your Windows 2003 installation disc, and click the Start button.
5. At this point, the install works like almost any other OS installation.

Using VMware Converter to Create a Virtual Machine

There may be times when you need to convert an existing physical image to a virtual machine; there are tools available for this. One use for this technology is to convert existing physical systems into virtual machines.

NOTE If you are like me the chances are good that you have an old Windows XP laptop or desktop system lying around that you are no longer using. If so, why not convert it to a virtual machine and use it for testing? It will cost you nothing more than a little time to convert it.

One of the easiest ways to create a virtual machine is to convert an existing physical computer to a virtual image. A tool for doing this is VMware vCenter Converter. You can download it from https://my.vmware.com/web/vmware/info/slug/infrastructure_operations_management/vmware_vcenter_converter_standalone/5_5.

The following steps will walk you through the process of using VMware to convert a physical image to a virtual machine:

1. Start the converter program.
2. Enter the IP address or hostname of the system you would like to convert.

3. Click Next once a connection is made.
A screen opens, prompting you to install the Converter Client Agent.
4. Choose the destination to which you would like to store the newly created VMware image.
5. Allow the process to finish. This may require some time if the image is large.

Once completed, you will have successfully created a VMware image.

Exploring Other Operating System Options

If you have decided to use VirtualBox instead of VMware Workstation, one of the benefits is that VirtualBox is able to set up VirtualBox images. Check out <https://virtualboximages.com/Free.VirtualBox.VDI.Downloads> and explore some of the ready-to-use images that are available to download. See if you can find the following operating systems and list their version and description.

OS	VERSION/DESCRIPTION	SIZE
PC/OS10		
CentOS		
Ubuntu		
OpenBSD		
ReactOS		
Gentoo		
Debian		

NOTE If you find a VirtualBox appliance that you would like to use in VMware, you can always attempt to export the appliance and then import it into VMware. While the process does not work 100 percent of the time, it is worth a try.

Running Kali from VMware

This exercise will demonstrate how to load Kali from the Wiley website:

1. Locate the `kali.iso` file that Wiley has made available, and copy it onto the hard drive. A good place to save the `kali.iso` file is `my documents/my virtual machine/kali`.
2. From the VMware Workstation menu, choose New Virtual Machine. Allow the wizard to walk you through the choices, and select the defaults for each setting. On the Guest OS screen, choose Other Linux and name the virtual machine Kali.

3. When the wizard finishes, choose Edit Virtual Machine Settings. Select Use ISO image, and browse to the `kali.iso` file. Then click OK.
 4. From VMware Workstation, select Start This Virtual Image. Kali should proceed to load.
 5. After Kali loads, you are ready to start using your new virtual machine.
- Congratulations: you now have Kali installed and running!

Installing Tools on Your Windows Virtual Machine

This exercise will discuss some of the tools you should consider installing on your Windows virtual machine (VM). If you have completed all of the previous exercises, you now have several Linux and Windows VMs. While Linux VMs such as Kali and Fedora Security Spin have all the tools you need installed, your Windows systems do not.

NOTE If you're trying to build the ideal lab environment, you won't be running or installing any additional tools on your base lab system. All tools and applications will be run from a virtual system. If you follow this approach, you will reduce the chance of something going wrong with your base laptop or desktop system.

OS	DOWNLOAD LOCATION
Wireshark	https://www.wireshark.org/download.html
NetworkMiner	http://sourceforge.net/projects/networkminer/files/latest/download
NetWitness	www.emc.com/security/security-analytics/security-analytics.htm#!freeware
Nmap	http://nmap.org/download.html
Cain & Abel	www.oxid.it
SuperScan	www.mcafee.com/us/downloads/free-tools/superscan.aspx
FOCA	www.pcadvisor.co.uk/downloads/3249362/foca-free-261/

NOTE You will be testing many of these tools in subsequent chapters. Setting everything up now will allow you to focus on using the tools and not having to install them later.