

SEC401 | SECURITY ESSENTIALS BOOTCAMP STYLE

401.I

Networking Concepts

SANS
Institute

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

Copyright © 2016, The SANS Institute. All rights reserved. The entire contents of this publication are the property of the SANS Institute.

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND THE SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, the SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by the SANS Institute to the User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between The SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO THE SANS INSTITUTE, AND THAT THE SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND), SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to the SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of the SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of the SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

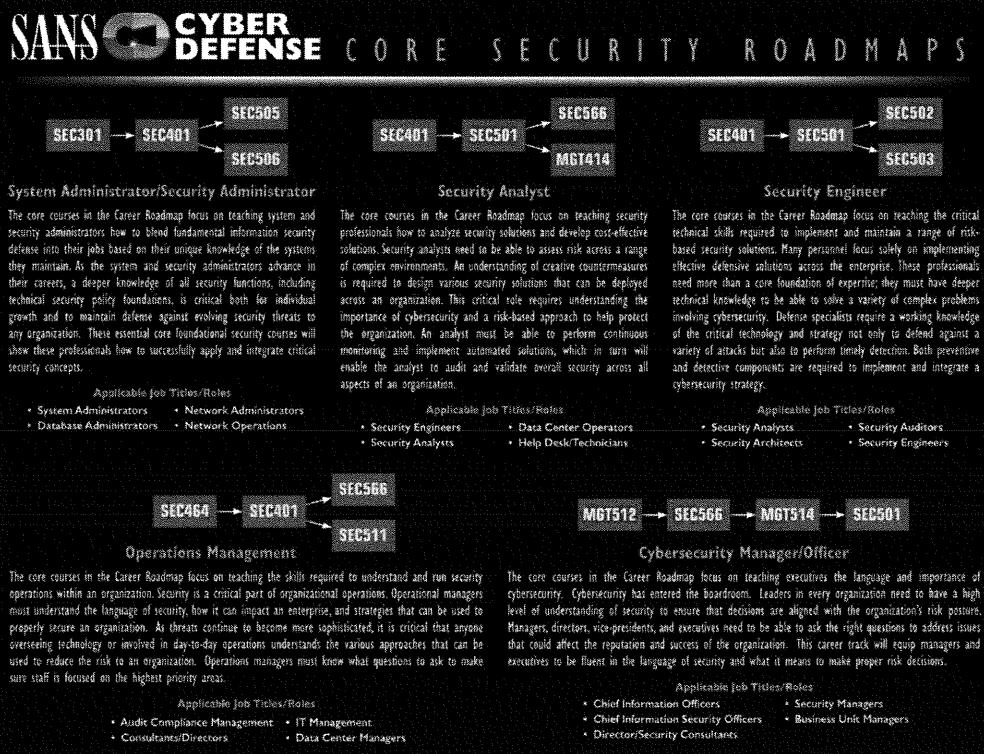
Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

SECURITY 401

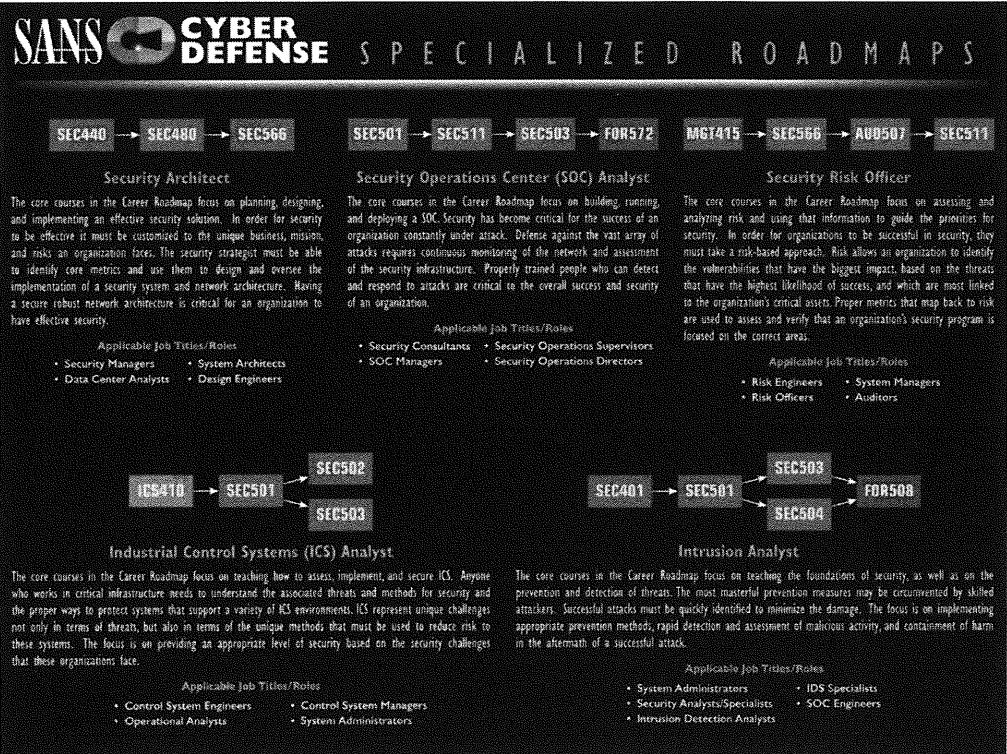
SANS Security Essentials

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

This page intentionally left blank.



This page intentionally left blank.



This page intentionally left blank.

SEC401 Day 1

- Setting up a lab and virtual machines
- Protocol stacks and IP concepts
- TCP, UDP, and ICMP
- Protocol analysis
- Wireless network security
- Labs:
 - Setting up virtual machines
 - Windows and Linux tutorial
 - tcpdump and decoding

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

This page intentionally left blank.

Module 1: Setting Up a Lab and Virtual Machines

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 1: Setting up a Lab and Virtual Machines

This section intentionally left blank.

SEC401 Labs

SEC401 requires that a student have a working copy of Windows 8 and Kali Linux installed.

Recommendation:

Windows 8 as a host operating system

Kali Linux as a guest operating system

Note: MAC users can run Windows 8 and Kali on two separate virtual machines.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

This page intentionally left blank.

Virtual Machines

- Allow software to run virtually on the same hardware:
 - OS-level virtual machines
 - Application-level virtual machines
- Host OS
- Guest OS

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Virtual Machines

Introduction

Even though the price of hardware is coming down every day, there are still many reasons why you would not want to purchase more of it. In addition to hardware's less expensive price tag, processors are becoming more powerful and memory is less expensive. Therefore, many companies have high-end servers and use only a small percentage of their processing capabilities. Running several pieces of software virtually on the same hardware has numerous benefits.

OS-Level Virtual Machines

The most common form of virtual machine is an operating system (OS) virtual machine. The OS virtual machine enables multiple operating systems to run independently on the same hardware. You can use a virtual machine as a laptop that needs to run different operating systems or as high-end server farms that need to run many applications at a reduced cost in hardware. The first use is more of a novelty and does not have huge cost savings to an organization. However, it is still a powerful tool for a network security professional who can run Unix tools on an XP or Vista laptop without having to perform a reinstallation of any components. This is one of the main reasons we cover virtual machines in this section. The modern security professional must have access to both Windows and Unix operating systems because there are some tools that will run on only one OS. Dual-boot machines require reinstallation of key components. Multiple laptops require additional hardware, and bootable CDs have limited read-only capabilities. Virtual machines are a perfect solution for solving this problem.

Application-Level Virtual Machines

As applications become more powerful, attackers are taking advantage of the increased functionalities to find vulnerabilities in the software. Removing these applications is not always an option, yet they are a prime source of malicious code that can infect an entire system. An exploit in a web browser can be used to compromise the entire application. One solution is to run application-level virtual machines. Now an application runs in a separate virtual machine so if it is exploited or compromised, the scope of the attacker is limited to a separate system and cannot compromise the main operating system in which all of the other applications run.

Host and Guest Systems

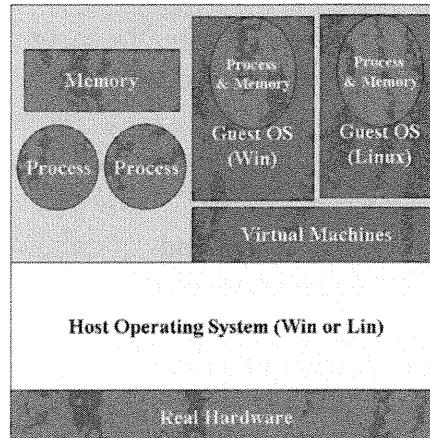
When talking about virtual machines, it is important to be able to distinguish between the main operating system and the virtual software. Because a computer needs an operating system to boot up the computer, it is referred to as the host operating system. On the host operating system, you would install a virtual machine application such as VMware. This virtual machine software enables you to run multiple guest operating systems that are actually applications running on the host OS. However, the virtual machine software segments out memory and hardware so they look and act like independent OSes, even though there is always one host and one or more guest operating systems at any given time.

Summary

As people recognize the benefits of virtual machines, they are going to continue to grow in popularity. From security professionals running multiple OSes on a laptop, to large corporations saving millions of dollars on hardware, the benefits are obvious. From a network security standpoint, it is critical to understand the value and benefit of virtual machines.

How do Virtual Machines Work?

- A computer needs a standard OS installed
- Virtual machine software is installed as an application
- Guest OSes are installed with the virtual machine software
- Virtual machine software is responsible for segmenting and creating virtual hardware



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

How do Virtual Machines Work?

Introduction

Virtual machine software (VM software) is a simple emulator for a computer, all created in software. You install a VM program on top of another operating system, known as the host operating system, such as Windows or Linux. Then, you can boot up virtual computers in the VM software, each of which is referred to as a guest operating system or a virtual machine. Each guest OS has its own memory allocation, virtualized network adapters, hard drive(s), and other hardware components. The different guests and the host appear to be truly independent operating systems, all running on the same hardware.

VM Software

VM software is loaded on the host operating system, just like any other application. After it is installed, its job is to interface with all of the hardware components and create virtualized hardware environments so other operating systems and applications can be installed. The operating system or application is not aware that it is running on a virtual machine. Therefore, you can install four different guest operating systems, each with a virtualized network interface card. This means each one can be assigned a unique IP address and connect and interface as if it was installed on four different computer systems and connected with a switch.

Virtualization

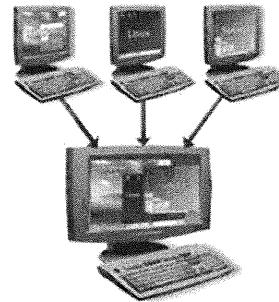
Virtualization is the workhorse of VM software. The VM software has to take the physical memory, process, NIC, and other hardware components and create virtualized components for the different guest operating systems to interface with. The VM software has to track the exchange between the physical and virtual components to make sure everything works properly.

Summary

When troubleshooting an operating system, it is always important to remember whether it is a host or guest operating system. Although they seem to work the same way, there are subtle differences between the two. A host operating system accesses the hardware directly and the guest operating system accesses virtual hardware through an emulator.

Benefits and Uses

- Main benefits are:
 - Multiple OSes on the same system
 - Server consolidation
 - Isolation of key components
 - Rebuild systems quickly
- Main uses are:
 - Security training
 - Incident response
 - Malicious code analysis
 - Digital forensics
 - Ethical hacking
 - Virtual security lab



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Benefits and Uses

Introduction

Virtual machines have had tremendous growth because they were commercialized over 5 years ago. In the early days, when hardware was very slow and expensive, you could barely run one operating system on a computer and a few applications without having performance issues. Starting in 2000, processors became very fast, and many systems had dual-core processors. In addition, it was not uncommon to have a minimum of 2GB of memory, even in laptops. This meant that on the average system, the processor and memory utilization were under 30%, which means plenty of power was available. This extra capacity could easily be used to run one or several virtualized operating systems.

Benefits

One of the main and most obvious benefits is being able to run multiple operating systems on the same computer. Whether you are bringing a laptop to a client and need to run tools on Windows and Unix, or you are a MAC user who wants to be able to access applications that are built for Windows operating systems, virtual machines enable you to use one computer to run many operating systems at the same time. In the past, you would have to either bring multiple computers or set up a dual-boot system. Neither one of these is a great option. From a server perspective, instead of having to buy a large number of systems, a small number of higher-end systems can run the same number of operating systems with less hardware to maintain.

From a Disaster Recovery Plan (DRP) perspective, instead of having ten servers with one OS and no redundancy on each, you could have two systems with five operating systems and four failover systems each and still have less hardware than before. If a system goes down, you do not have to rebuild the system from CDs or restore from a backup. You would just load the guest image and get the system running in minutes. From a security perspective, key components can be isolated and contained to reduce or limit exposure.

Uses

With the main benefits being ease of use and portability, the uses of virtual machines are almost endless. Starting with the most obvious, virtual machines are great for training. For example, you can bring a single computer and run multiple operating systems that are pre-built by the instructor. Incident response, malicious code analysis, and digital forensics all allow a security professional to have a portable security lab on a single computer with all of the tools and software they need at their fingertips.

Summary

Virtual machines benefit the individual user and the high-end data center. Virtual machines enable security professionals to access a wide range of operating systems without having to purchase expensive hardware. This enables us to create virtual security labs with minimal effort. In the early 90s, if you wanted an effective lab, you had to buy several computer systems with removable drives. Now, with a few systems and virtual machines, you can simulate an entire corporate network with minimal expense.

Types of Virtual Machines

- VMware *
 - VMware Player
 - VMware Workstation
 - VMware Server
 - VMware Fusion
- Virtual PC
- Parallel
- Oracle VirtualBox
- Many more ...

* Coverage in class

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Types of Virtual Machines

Introduction

Whenever something becomes popular, several companies will produce competing products, giving the consumer a wide range of choices. If you want to buy a mid-size car, there are many makes and models to choose from. Based on the popularity of virtual machines, it should not surprise you that you can choose many products.

Vendors

If you search on Google, you can quickly see a long list of vendors. Microsoft offers Virtual PC and you have parallel choice for the Macintosh computer system. There is also a wide range of products for Windows and Linux, starting with VMware.

VMware

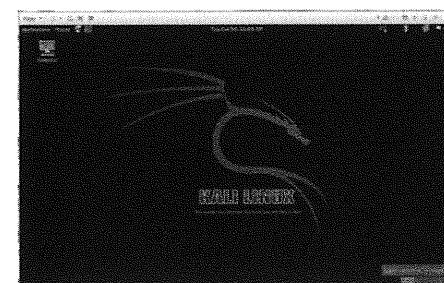
Because of the popularity of VMware, we focus on it for this class. Not only is it widely available, but many people are also familiar with it. VMware Player is free of charge, which enables anyone to run virtual machines. VMware Workstation is the commercial version of VMWare Player that has additional features and capabilities. There is also VMware Fusion for operating systems such as the MAC and, of course, for high-end data centers, a wide range of server products is available.

Summary

We are going to use VMware for this class, but if you use a different virtual machine product, the guest operating systems will still work the same way. Just as some people like BMW, whereas others like Mercedes or Lexus, there is no right or wrong answer when it comes to virtual machines.

VMware Player

- Free
- Runs on Windows and Linux
- Allows virtual machines to run
- Less functionality than commercial solutions
- Works well for virtual lab environments



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VMware Player

Introduction

With most popular tools, free versions of a tool exist, but as they become popular, commercial companies create separate products with more features and a different interface than the free counterpart. However, with VMware, the company realized that people would want to try new features and run virtual machines without having to buy a commercial product. Therefore, they created VMware Player, which is a free version of VMware that is fully compatible with the commercial versions. This means you can take virtual machines created with any other VMware product and run them in VMware Player. However, because the product is free, it has limited functionality.

Benefits

One of the big reasons for VMware Player is it gives people an easy way to evaluate virtual appliances, which are pre-built OSes and applications that someone can run with minimal overhead. For students, or even security professionals, it provides a simplified method of being able to run different operating systems and tools with no additional cost for software. It also provides a great way to create virtualized network environments. With VMware Player, you can install multiple operating systems and have them communicate through network protocols as if they were on separate networked machines. This enables you to simulate a lab environment with only one system.

Features

Some of the key features of VMware Player are:

- The capability to run 32- and 64-bit operating systems from a standard 32-bit OS. For example, even if you have a standard 32-bit operating system installed, through the virtualization features of VMware Player, you can still run 64-bit operating systems.
- The capabilities of multiple CPUs to support more efficiently run virtual machines.

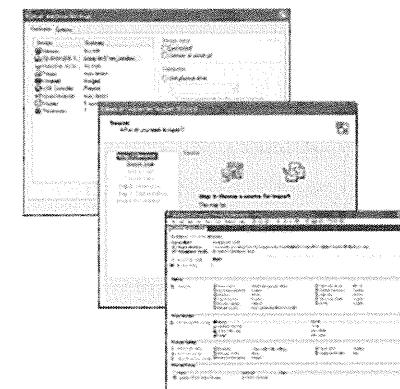
- Ease of backup and portability of operating systems. For example, an instructor can provide an entire operating system image on CD to all of his students and guarantee that everyone has the same environment and tools installed. In addition, if a student has problems or issues, he can quickly reload the original image.
- The capability to test new devices. Because VMware Player has full support for USB devices, you can test and run new devices in a virtual machine, minimizing the chances of damaging your host OS.

Summary

Even with its scaled back capability over commercial VMware solutions, VMware Player provides a simple, easy, free way for you to test and run virtual machines. As you will see, there are many benefits and reasons for using one of the commercial solutions, but VMware Player lets you “try before you buy.” This enables you to test and understand how virtual machines work without having to spend any money on the virtual machine product.

VMware Workstation

- Commercial desktop VMware solution:
 - 30-day free trial
 - Upgrade path from VMware Player
- Benefits:
 - Capability to create snapshots and cloning
 - The ability to create true multi-tier environments
 - Secure portability of virtual machines



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VMware Workstation

Introduction

In order to embrace all of the capabilities of virtual machines for a workstation, VMware created VMware Workstation. In addition to all of the base functionality that is provided in Player, Workstation provides increased support and capability to take full advantage of virtual machines. If you are using VMware Player and it does not have all of the configuration options you need, but you are not sure if you want to purchase VMware Workstation, you can use a free 30-day trial to better understand the enhanced functionalities and determine whether it is worth the cost to purchase the full version.

VMware Workstation

In addition to all of the features of VMware Player, Workstation provides additional features that can be used to perform full testing of new applications and patch management. In many cases, people are not sure whether removing a feature or patching a system will cause interoperability problems. Now you can take a snapshot of an existing system, run it in a virtual machine, install patches, and make changes to determine the stability of the system prior to putting it into production. VMware Workstation is a better, more enhanced version of Player but with additional features and capabilities.

Features

VMware Workstation has many new capabilities and benefits. Some of the most critical benefits are:

- Full interaction and capability to work with virtual machines. You can run multiple virtual machines and drag and drop files between the different operating systems, providing seamless integration. In addition to creating virtual machines from scratch, you can also take an existing physical machine and convert it into a virtual machine.

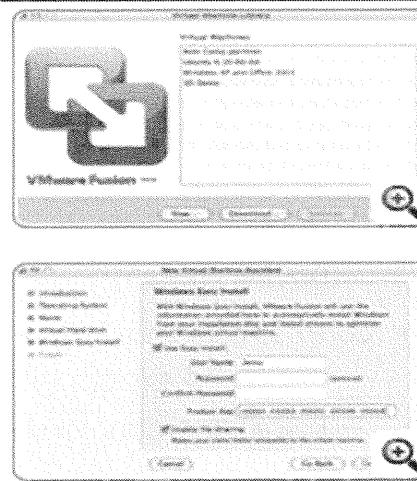
- The capability to create snapshots of a virtual machine at any point. During testing and patching, you make changes to a system that can impact the functionality of the system. If you make multiple changes and one of the five changes breaks something, you do not want to have to start from scratch. Snapshot allows you to create snapshots of the system at different points to make it straightforward to return to a prior point in time, within the same virtual machine. In addition, you can also create videos showing all of the changes that have been made and exactly what was done, which is helpful for not only training purposes, but also for forensic work.
- The capability to create multi-tiered environments. With VMware Player, if you want to run five different systems, you have to run five different independent virtual machines. With multi-tier support on a single virtual machine, you can have multiple operating systems that can simulate a true environment with firewalls, servers, and back-end systems.
- The capability to clone environments so that you do not have to start from scratch each time you want to work with a virtual machine.
- Secure portability of virtual machines through ACE. Often virtual machines have sensitive data or information in them, and if they are put on USB or other portable hard drives, there is a risk of compromise. With the ACE feature, you can now securely put virtual machines on portable devices, reducing the risk of compromise.

Summary

While VMware Player has basic functionality, if you are going to use virtual machines as a key part of your organization, the extra benefits of VMware Workstation are worth the investment. Being able to control changes, creating multi-tiered environments, and providing security to protect the portability of virtual machines are important features that are needed in most environments.

VMware Fusion

- Runs other operating systems on a MAC
- Integrates seamlessly into the Mac OS with Unity
- Features:
 - Interface across portable devices
 - On-the-fly virtual machine creation
 - Access other OSes with no reboots



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VMware Fusion

Introduction

For many reasons, more and more people are starting to use Macintoshes. However, one of the main problems is that many applications do not run on a MAC, which limits the value of the operating system. With VMware Fusion, you can run other operating systems as virtual machines, which enables you to run non-MAC applications on your system. This provides many of the benefits that you get with VMware Workstation and virtual machines for the MAC operating system.

Unity

Virtual machines have benefit and value, but you are still running a separate operating system in a separate window, which does not provide seamless integration. Unity allows you to integrate a Windows application into the MAC operating system. Instead of having to run a separate virtual machine window, the key applications work alongside the native MAC applications, allowing for drag and drop features and integration with the base system functions. This means that Windows applications will appear in the Application menu and are accessed just like any native application. In addition, it provides full support for Vista 32- and 64-bit operating system applications.

Features

VMware Fusion brings all of the benefits of virtual machines to the MAC. Some unique features are:

- Enhanced graphic interaction with the operating system. One of the many benefits of a MAC is the enhanced graphic capability. This is now available for applications on other operating systems.
- Capability to interface external devices with applications in different operating systems. For example, a PDA device can interface directly with Outlook running under Windows.
- Better interface between documents on the host operating system and documents on the guest operating system.

Summary

With the power of virtual machines, you can pick the operating system you want and have critical applications from other operating systems integrate into the core functionality of the system. VMware Fusion enables a MAC to run both Windows and Linux applications.

Installing and Setting up Virtual Machines

- Virtual machines and VMware are installed like any other applications
- Setting up VMware requires understanding the hardware on the system:
 - Memory allocation
 - Device support
- Load virtual machine images:
 - Unzip and extract ISO if provided by someone else
 - Install operating system from original media

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Installing and Setting up Virtual Machines

Introduction

In order to harness the power of virtual machines, there are two distinct components that need to be installed: the virtual machine software and the OS/application images. The virtual machine software, i.e. VMware, is an application that is installed and responsible for interfacing with the host OS and the hardware to create a virtualized environment that other software can run in. Once this environment is set up, software or applications need to be installed to run in the virtual environment.

Installation

VMware Player and Workstation are just like any other application that gets installed on a host operating system. You would either download the software (required for Player or the 30-day free trial for workstation) or you can purchase shrink-wrapped software and install from CD. The software has an installation wizard that will walk you through the process. Since Player is a scaled back free version, it has less screens and options when installing. Once the VMware software is on your system, you then have to configure it prior to loading software images on the system.

Setting Up and Configuration

With VMware Player there is very little to setup. When you run the software, it automatically asks you for the image you want to load. If you do not specify an image, it exits out of the software.

Once an image is loaded, there are basic configuration options, but it is really just meant to run images with little configuration support, which as you will see is both a positive and a negative. It is simpler to use but you have less control over the environment. It is important to note that even though Player has less options in the interface, with both Player and Workstation you can go in to the actual configuration files and edit those by hand. That is usually not recommended, since if you make mistakes you could cause the image to no longer work. VMware Workstation has many more controls in terms of how the system runs and how the images interface with the environment.

This will be covered later in this module.

Loading Images

After your VMware software is installed, you need to load images to get to the true power of the software. The images are the actual operating systems or applications that run in a virtual environment and can be created in several ways. First, you can install the software, just like you would a real operating system. For example, put the Windows or Linux CD in your system and it boots up in the virtual window and installs the software. Then, obtain a pre-configured image either from a removable drive, or, because they are very large, in the form of a zip or ISO image. When this is extracted to a directory, point your VMware software to the directory and it will run.

Summary

Installing and setting up virtual machines is a straightforward process. Because it is so easy, it is one of the many reasons virtual machines are so popular. In a matter of minutes, you can have a new operating system running on your box or test out a new application. Instead of having multiple machines or constantly having to rebuild a machine, virtualization allows you to create new development and production environments very quickly.

VMware Tools

- Not installed by default.
- VMware tools must be installed in the guest operating system
- Key benefits:
 - Increased performance
 - More control over screen resolution
 - Provides better interaction with the guest operating systems
- Installed differently depending on the host operating system
- Included within the VMware ISO



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VMware Tools

Introduction

With almost any piece of software that you install today, there are always updates or enhancements that the vendor builds after the software has been released. If you are running XP or any Windows operating system, it is critical to make sure you install the latest service pack, which contains updates and enhancements. VMware is no different. VMware Workstation has an update called VMware Tools. It is provided automatically in the ISO, but it is not installed by default.

VMware Tools

VMware Tools can be thought of as an upgrade to the existing VMware virtual machine software. However, it is an enhancement to how the guest operating system interacts with the host and the hardware, and it must be installed on each guest operating system you use with VMware. The good news is that it needs to be installed only once for each guest operating system, not every time the guest is started.

Benefits

The VMware Tools package gets installed inside a virtual machine that allows VMware to better interact with the guest operating system. By including hooks in the guest OS, VMware Tools can improve performance and give you better control over the guest system's GUI. You are also better able to set resolution and colors once VMware Tools is installed.

Typical VMware Setup

- Host operating system:
 - Windows 7/Windows 8
 - VMWare Player or Workstation
 - VMWare Tools
- Guest operating systems:
 - Linux/Kali
- Optional setup:
 - Run a second version of Windows as a guest
 - Host version of Windows is not used for any tools

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Typical VMware Setup

Introduction

You can utilize and harness the power of virtual machines in many ways, depending on personal preference, the job, or other requirements. However, to baseline the process for this class and other classes, we are going to define a standard VMware setup. We'll use this setup in this class. If you want to use a different setup, you can, but this one provides a common baseline for this module and other follow-on classes. As a starting point, we will use VMware as our virtual machine software.

Host Operating System

Most laptops or desktop devices in which you are going to run virtual machines have a Windows operating system installed by default. In addition, many corporations require the use of Windows for desktop systems. Therefore, it is a natural choice to use it as the host operating system. Because you'll often want to add virtual machines to an existing environment, this configuration enables you to simply install a new application on your system without requiring you to reinstall other applications or make changes to existing systems. However, it is important to remember that one of the benefits of virtual machines is the capability to easily reload the operating system. If you are running tools on the host operating system and you crash the system, it is much harder to recover than if it was running on a guest operating system.

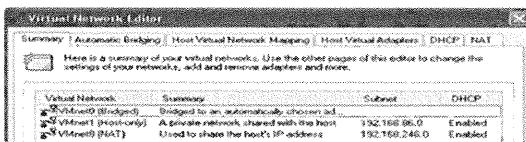
Guest Operating System

If you work in security, you need to have access to both Windows and Unix/Linux operating systems. Some tools run only on Windows, and some tools run only on Linux. Therefore, you need to know how to use both operating systems. The original solution to this was dual-boot systems or bootable CDs, but the main problem is that you can run only one operating system at any given time. Ideally, you would run both operating systems simultaneously. Because Windows is installed on most systems automatically, it makes sense to run a variant of Linux as a guest operating system. In addition, because Linux is free, you can freely distribute Linux images with all of the tools installed. The tools can be distributed to people who can foot the guest operating system and who can access the tools without being required to install or compile programs. For many classes, the instructor will give you a Linux image that you can run as a guest in VMware with all of the tools needed for the class precompiled.

VMware Network Options

- Virtual machines can use one of three network options:
 - Host-only network.* Nothing other than the host operating system can get to the virtual machine across the network.
 - Bridged network.* The host and virtual machines behave as though they are sitting next to each other on a switch.
 - NAT. The host acts as a NAT device, which the virtual machines sit behind.

* Two main methods used



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VMware Network Options

Introduction

One of the many benefits of running virtual machines is the ability to connect between different operating systems. For example, you might want to create a virtual lab where the host and guest operating systems can communicate only on the local system, but act as if they are connected through a hub. Or, you might want your guest operating system to have its own separate IP address and access the network independently of the host operating system. The good news is that VMware gives you plenty of options for setting up and controlling networking on your system.

Network Options

Virtual machines can use one of three network options:

- Host-only network:** With this option, nothing other than the host operating system can communicate with the virtual machine.
- Bridged network:** With this configuration, the host and virtual machines behave as though they are sitting next to each other on a switch. This introduces the virtual machine MAC address on the LAN. Also, it puts the host network interface in promiscuous mode (to capture traffic destined for the virtual machines, the host will have to grab packets destined for MAC addresses that don't match the hardware address).
- NAT:** In this mode, the host acts as a NAT device, which the virtual machines sit behind. All packets get their source IP translated so that they appear to have come from the host instead of the guest operating system.

The two most common methods you will use are host-only and bridged networking.

Summary

By harnessing the full power of virtual machines, you can run various security tools across different operating systems and test them both locally and across the network. VMware gives you full flexibility for running a range of network options on your local system.

VMware Summary

- VMware is very powerful:
 - Supports many operating systems at both the host and guest level
 - Serves as a key application for the security professional
 - Provides value in data centers for server consolidation
- To understand how to configure and use it, pay close attention to the network settings

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VMware Summary

Introduction

One of the many challenges of network security is that it is always changing. As a security professional, you need to be able to have a lab that you can use to test out the latest tools and understand how attackers are breaking into systems. In the past you had to buy multiple systems with removable hard drives and constantly re-install your operating systems when they crash. Now, with the power of virtual machines, it becomes much easier to build a virtual lab.

Power of Virtual Machines

Although there are many benefits to virtual machines, the two key aspects from a security professional's perspective is the ability to rebuild a system very quickly and set up multiple virtual environments. During the testing of systems or while performing forensics you can potentially put a system in an unstable state. On a normal system, to reinstall the entire operating system and all of the applications requires a time commitment. With virtual machines, you can create a clone or even a snapshot, so if there is a problem or issue, you can revert back to the original image very quickly. The second benefit is you can run a virtual lab on a single computer. You can run four different operating systems in a sample configuration for your company and perform tests on all of them from a single portable system.

Summary

Beyond being a valuable tool for a security professional, virtual machines have many benefits to an organization. From virtualization in a data center to reduce costs, to use on a desktop to minimize the impact of a security breach, virtual machines are critical components of the modern organization. However, it is important to remember that there is no perfect solution. Although virtual machines have many benefits, it is important to test them thoroughly to make sure there are no unknown security breaches.

Module 2: Network Fundamentals

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 2: Network Fundamentals

This section intentionally left blank.

Network Fundamentals

SANS Security Essentials I: Networking Concepts

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Network Fundamentals

Introduction

In order to understand principles in network security, you have to have a core understanding of networking. We strive to provide you with knowledge and skills that are essential for carrying out security responsibilities in your organization. The first few modules lay the groundwork for specialized security topics that are covered later on. We begin this course with the basics of network operation and design and then proceed to discussing particulars of the TCP/IP protocol suite. We show you how to analyze network traffic using a sniffer. We also take a close look at virtual machines and how they can be used by a security professional. We conclude this section of the book and course by focusing on physical security. These concepts are crucial to security because the network offers the backdrop for performing business activities. Coincidentally, the network is also the medium for attacking the very systems that make legitimate activities possible. To be able to protect the network, you must first understand how it operates.

Objectives

- Network Principles
- Network Hardware
- Network Design

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

This module discusses fundamental principles you need to know to build a secure network. We review several Local Area Network (LAN) topologies—bus, ring, and star—and we explain how they relate to low-level communication protocols, such as Ethernet, Token Ring, and Asynchronous Transfer Mode (ATM). Next, we look at network devices that you are likely to find on the network: switches and routers. We conclude the module by putting all of the pieces together and looking at network design.

Network Principles

The student will understand and
be able to identify the different
types of networks
and the most common network
technologies in use today.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

This page intentionally left blank.

Types of Networks

- LAN (Local Area Network)
- WAN (Wide Area Network)
- MAN (Metropolitan Area Network)
- Internet
- PAN (Personal Area Network)
- NAN (Neighborhood Area Network)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Types of Networks

A key tenet of network security is to know thy system. You cannot secure something that you do not understand or know how it works. To be proficient in network security, you have to understand the different types of networks because each network type poses different challenges, issues, and risks.

Local Area Network

A Local Area Network (LAN) is a relatively small network that is confined to a small geographic area, such as a single office or a building. Laptops, desktops, servers, printers, and other networked devices that make up a LAN are located relatively close to each other. A key characteristic is that all of the equipment that comprises a LAN is owned by a single entity. [1]

Threats Posed to a Local Area Network

From a security context, LANs are the point at which trusted users typically access your network and server resources. Often, enterprises extend too much trust to users in LANs who have otherwise unrestricted access to information resources. Consider the plight of an organization that fires an employee, but permits the employee to access the network under the guise of removing personal data. With unrestricted access to network resources, the disgruntled employee has the ability to delete or tamper with information that is critical to the organization. Even happy, trustworthy employees can be a critical threat to information security. An employee who is tricked into installing malicious software or accidentally introduces a computer virus or worm to an organization can cause immeasurable damage if he is granted access to critical systems. Once someone has access to information, he potentially represents an insider threat.

For example, an employee of a large corporation logged into his computer and set off a logic-bomb that deleted all the programs that ran the company's engineering operations. A former system administrator turned disgruntled employee, who had been fired from the company shortly before implementing his attack, planted the logic-bomb. The result: The company lost \$12 million in revenue and had to lay off 80 employees as a result of the losses. [2]

It is easy to identify employees as the potential inside threat, with all others being in the external threat category. The problem with this classification method is that LAN users are not always employees. Contractors, business partners,

vendors, and students are all examples of people who might use a company LAN but are not trusted with limitless access to information resources. It is important to consider all access to LAN resources—not just traditional users—when evaluating the internal threat to an organization.

Metropolitan Area Network

The term Metropolitan Area Network (MAN) is typically used to describe a network that spans a citywide area or a town. MANs are larger than traditional LANs and predominantly use high-speed media, such as fiber optic cable, for their backbones. MANs are common in organizations that need to connect several smaller facilities together for information sharing. This is often the case for hospitals that need to connect treatment facilities, outpatient facilities, doctors' offices, labs, and research offices for access to centralized patient and treatment information. MANs share many of the same security threats as LANs, but on a larger scale. The plight of an administrator in a central location, granting access to countless offices that are scattered within a city, is a difficult one. This plight demands strict access control mechanisms to protect against unauthorized information access.

Wide Area Network

A Wide Area Network (WAN) covers a significantly larger geographic area than LANs or MANs. A WAN uses public networks, telephone lines, and leased lines to tie together smaller networks such as LANs and MANs over a geographically dispersed area. Connecting devices in different geographic areas together for information sharing make WANs an important piece of enterprise networks.

The Internet

The Internet is an example of a network that connects many WANs, MANs, and LANs into the world's largest global network. Internet Service Providers (ISPs) are responsible for maintaining the integrity of the Internet while providing connectivity between WANs, MANs, and LANs throughout the world. ISPs provide customers access to the Internet through the use of points-of-presence (POP), also called network access points (NAP), in cities throughout the world. Customers provision access to POPs from their own WANs, MANs, and LANs, giving Internet access to their users.

In addition to providing customer access to the Internet, ISPs also provide connectivity between each other at "peering points." Large peering points are called metropolitan area exchanges (MAE). The acronym MAE is pronounced as "May." ISPs are able to exchange traffic originating in one ISP that is to be delivered to a different ISP.

Personal Area Network

A more recent term used to describe a type of network is a Personal Area Network (PAN). PAN networks are usually wireless and are established in an on-demand or ad-hoc fashion when needed to communicate between two or more devices. PAN networks can be used between devices owned by two different parties or between two devices owned by one person, such as a PDA and a laptop or mobile phone. These networks are usually characterized as short-range and are often limited to 10 meters or less in range.

An example of a PAN technology is Bluetooth wireless networking. Bluetooth is designed as a cable-replacement technology, allowing users to discard the serial and USB cables used by many of today's peripheral devices and rely on a Bluetooth PAN for communication. Bluetooth PAN's support of up to seven devices in a single network and can be used for proprietary protocols (such as PDA synchronization) or standards-based protocols, including Internet access over IP and the Bluetooth Network Encapsulation Protocol (BNEP).

Summary

It is critical to understand the type of network you are dealing with because each has its own unique set of challenges and risks that need. Too often organizations try to implement security when they do not understand the core concepts of the networks they are trying to secure.

References

- [1] <http://comm.ncifcrf.gov/networking/whatislan.html>
- [2] http://www.secretservice.gov/ntac_its.shtml

Physical and Logical Topologies

- Physical topologies:
 - How the network is actually connected
 - How the data actually flows
 - Wired or wireless
 - Verification of physical topology is critical to ensure security
 - Star topology most common
- Logical topologies:
 - How you communicate across the wires
 - The meaning of the information
 - Language
 - Ethernet most common (CSMA/CD)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Physical and Logical Topologies

There are two types of topologies that together describe how workstations on a network are able to communicate. To secure a network, you have to understand how they are physically connected and how they communicate.

Physical Topologies

A physical topology describes how the network is wired together. It is the layout of how systems are connected via cables or wireless devices. Wire-based physical topologies are easy to visualize because they are interconnected according to simple geometric patterns.

Logical Topologies

After the systems have been interconnected, they must know the rules for sending signals to each other. These rules are specified by media access protocols. Some examples of logical topologies are:

- Ethernet
- Token Ring

These protocols are responsible for making sure that a signal sent by a system finds its way to its destination. The process that the protocol follows to send data over the cable, regardless of how it is physically wired, can be described using a *logical topology*.

Comparison of Physical and Logical Topologies

Physical and logical topologies are generally independent of each other. As you will see, a Token Ring network—which uses a logical ring topology—is usually wired according to a physical star topology. Often, there is a relationship between a physical and a logical topology that results in some pairings being used more often than others.

To better understand the distinction between physical and logical topologies, consider how humans communicate. In most cases, our verbal interactions are guided by the grammar of a particular language, such as English. The English language has numerous rules that dictate how we should form words and sentences to help provide meaning to what we say. English grammar, then, is our logical topology, which describes our communication protocols. The physical topology of human interactions defines the systems that we use to communicate. For example, a telephone is one such physical topology; postal mail is another. A single logical topology (English) can be used with multiple physical topologies (telephone and mail). Similarly, each communication system can act as a carrier for different human languages.

Summary

A network security professional should understand the physical topology that is in place and make sure it is implemented correctly. Changing one wire in a physical topology can greatly reduce or eliminate security, such as bypassing a firewall. Once the physical topology is understood, it is critical that you evaluate the logical topology that is in use and make sure it is properly secured.

Ethernet

- Ethernet is shared media:
 - CSMA/CD (carrier sense multiple access with collision detection)
- Most common logical topology or layer 2 protocol
- Steps taken to communicate:
 - Listen before transmitting
 - Make sure only one station is transmits at a time
 - Monitor transmissions to check for collisions

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Ethernet

Ethernet is the most popular media access protocol (layer 2 protocol) currently used on LANs. In fact, it is nearly ubiquitous for networking, with the exception of the backbone itself. A chunk of data transmitted by Ethernet over the wire is called a *frame*. On an Ethernet network, only a single node should transmit a frame at a time. If multiple systems transmit simultaneously, a collision occurs. This collision can cause both signals to fail and require the systems to retransmit their frames.

Collisions

To keep the number of collisions to a minimum, a system is required to check whether anyone else is already transmitting before placing a frame on the wire. If another system's signal is already on the wire, the system is expected to wait, according to the algorithm designed, to give each node a fair shot at using the network. If the line is clear, the system generates a signal and monitors the transmission to make sure that no collision occurred. These properties are summarized under Ethernet's designation as a Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocol.

Types of Ethernet

Ethernet specifications actually define more than just protocols for sending signals over the wire. Other properties include cabling requirements for transferring data at desired rates and the maximum length of the wire segment. In addition, Ethernet standards specify which physical topology should be used for a particular type of Ethernet communication.

Summary

Because Ethernet is the most common type of layer 2 protocol in use on networks, it is important to understand how it works. The shared segment aspect gives maximum flexibility, but it can also cause availability problems with collisions and confidentiality problems with sniffing the traffic.

WAN Technologies

- Dedicated lines:
 - T1 or T3 line
 - E1 or E3 line
 - Dark fiber
- MPLS (Multi Protocol Label Switching):
 - Unified data carrying service
 - Replacing Frame Relay and ATM
- Integrated Services Digital Network (ISDN):
 - Traditional phone lines
- DSL:
 - Traditional phone lines - Distance limitations
- Cable modems:
 - Cable TV lines
- FiOS:
 - Fiber network

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

WAN Technologies

Token Ring and Ethernet protocols are great for sending signals over relatively short distances; however, they are not effective at carrying data between geographically distant sites. The connections among these greater distances are better served by a WAN. In this section, we take a brief look at technologies designed for WAN communications.

Dedicated Lines

One way to interconnect two remote sites is to provision a dedicated point-to-point link between them. This link often takes the form of a high speed T1, T3, E1, or E3 line, which is leased from a carrier company. A dedicated line is used solely by the company that leases it and does not carry data for any other entity. As a result, the benefits of using a dedicated line to establish a WAN connection include predictable availability of bandwidth, reliability of the connection, and confidentiality of the transported data.

The primary disadvantage of a dedicated line is its cost, which is usually proportional to the distance between the two sites. The expense of leasing a T link that spans more than several miles can make a dedicated line a cost-prohibitive solution. This consideration is most relevant for companies seeking to establish point-to-point links between more than two sites in geographically dispersed regions. In such situations, using a packet-switched technology such as Frame Relay might be a better solution.

Frame Relay

Frame Relay is a WAN technology that is similar to Ethernet and Token Ring in that it is based on packet switching. This means that it breaks transported data into packets, with each packet traveling individually and, potentially, via a distinct path that is part of the same connection. The packet-switched nature of Frame Relay enables multiple companies to share the same WAN medium.

Unlike direct point-to-point links, Frame Relay is not priced according to the distance between sites but according to the amount of bandwidth used. Because multiple companies use the same Frame Relay medium, the cost of using

Frame Relay to interconnect sites is usually less than the cost of establishing dedicated lines. A company wishing to connect a new remote office to its global WAN network simply needs to pay for a link to a Frame Relay cloud, instead of provisioning dedicated lines to the other offices.

MPLS

A number of carriers use Multiprotocol Label Switching (MPLS). The advantage of MPLS is that it supports IP traffic, including standard networking IPv6, VoIP, and IP Video. MPLS is a packet-switching technology that provides a unified data carrying service. In the OSI protocol stack model, it is often considered a layer 2.5 technology because it sits between layer 2 and 3. MPLS provides similar technology and is seen as a replacement for Frame Relay and ATM because it better adapts to current network technology.

ISDN, DSL, and Cable Modems

ISDN, DSL, and cable modems all offer network connectivity to businesses and consumers through telephone companies, cable television companies, and Internet service providers. These services provide access to public and private networks for many organizations.

ISDN networks used a more traditional "dial-up" connection over telephone company networks to connect to remote networks. Using a service profile identifier (SPID), which is a 10-digit phone number and a 4-digit ISDN connection identifier, an ISDN network establishes a connection to other remote ISDN networks. In many cases, ISDN connections connect with any remote caller, potentially granting an attacker with an ISDN line to connect to other networks in an unauthorized manner. Because these ISDN lines are primarily intended for backup purposes, these "backdoor" connections might go unnoticed for months.

DSL

Digital Subscriber Line service is an offering from telephone companies to provide high-speed network access over traditional telephone lines. DSL is widely adopted by businesses and residential properties for access to public networks, such as the Internet. The low cost, the ability to use existing phone lines, and high-speed access are three factors that contribute to the widespread adoption of a DSL service.

The installation and configuration of DSL is easy. Because it operates over traditional telephone lines with a different modulation frequency, DSL service can be installed on existing telephone lines and used at the same time as traditional telephony service. For businesses, DSL threatens the integrity of perimeter defense systems. It is not uncommon for an organization to have rogue DSL lines installed and connected to their corporate networks as a means of bypassing access and firewall restrictions on primary Internet connections. These connections are typically unprotected, offering an attacker the ability to bypass traditional perimeter defenses.

Cable Modems

A third network connectivity alternative is available through cable companies. Directly competing with DSL service and telephone companies, cable television companies provide Internet access over their existing cable TV networks. Using the data over cable interface specification (DOCSIS), which was the International Telecommunications Union (ITU) approved in 1998, cable modem networks are the most prevalent form of high-speed network connectivity for residential users, with some adoption for businesses where available.

WAN Aggregation

As WANs increase in size and bandwidth options, it may become necessary to implement an aggregation device. These devices are able to aggregate multiple WAN data links and provide redundancy among multiple ISPs. They support most WAN protocols, such as PPP, HDLC, Frame Relay, as well as many types of WAN interfaces (T1 and ISDN).

Summary

There are many choices when determining a WAN solution for your home or business. Each one has associated functionality, cost, and security risks associated with them. However, with a proper network design, any of the WAN solutions can be properly secured.

Network Hardware

The student will understand network hardware components.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Network Hardware

This section intentionally left blank.

Network Devices

- Hub: Replicates traffic onto all ports, minimal security
- Bridge: Maintains track of network addresses, segments traffic, and breaks up collision domains
- Switch: Micro-segmentation with each port receiving traffic for the appropriate host using the MAC address
- Router: Forwards or drops traffic based on the destination IP address

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Network Devices

Several types of devices are commonly used at the core of the network to provide a reliable and flexible communication medium. It is important to understand how they function because each has inherent security strengths and weaknesses.

Hub

A hub operates by "repeating" data that it receives on one port to its other ports. As a result, a data frame transmitted by one system is retransmitted to all other systems connected to the hub. A classic hub does not have traffic-monitoring capabilities and cannot control which ports should or should not receive the frame, forming a large collision domain. This property of a hub has significant security implications because a system connected to the hub may be able to intercept a data frame destined for someone else.

Bridge

A bridge is used to connect two physical segments of a network in much the same way as an over-the-water bridge connects two sections of a road. When a bridge receives a data frame on one of its ports, it makes a decision about whether or not the data should be sent to the other port. This functionality allows a bridge to automatically control the flow of data between network segments that it connects.

In order to decide when to replicate frames from one port to another, the bridge learns which systems reside on which network segment. It accomplishes this by automatically recording the MAC addresses of frames that pass through it to construct a table that maps MAC addresses to network segments. If a bridge needs to process a frame destined to a MAC address that is not in the table, it forwards the frame anyway.

Switch

A network switch combines the functionality of a hub and a bridge into a single device. If you think of a switch as a bridge with more than two ports, you will get the idea. Like a hub, a switch can retransmit data to multiple ports. Additionally, an Ethernet switch keeps track of MAC addresses attached to each of its ports, which grants it the

traffic control capabilities of a bridge. By monitoring and controlling traffic between its ports, a switch will direct a data frame only to the system or network segment for which it is destined, narrowing each port to its own collision domain. Sniffing becomes very ineffective with switches.

Router

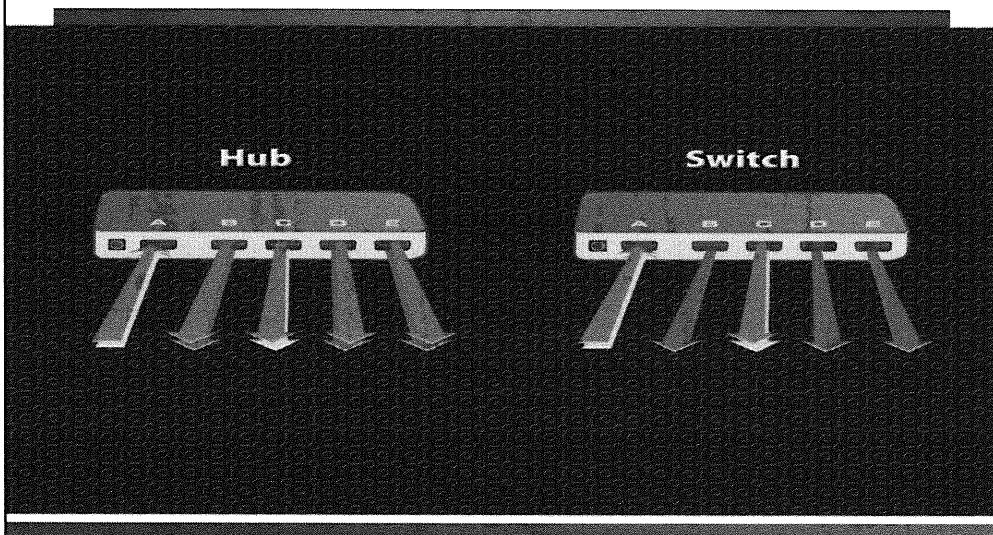
Routers are often considered to be perimeter devices because they interconnect logical networks. A switch or a bridge, on the other hand, connects physical segments that reside on the same logical network. Much of the Internet relies on routers for determining which paths packets should take to get from one network to another. Similar to a switch or a bridge, a router makes decisions about where to direct data that passes through it. A switch makes its decisions by tracking MAC addresses, whereas a router operates on a higher layer (Layer 3) by looking at IP addresses when forwarding packets.

Unlike a switch or a bridge, which transmit traffic to unknown destinations, a router drops traffic if it does not know where to send it. Routers need to be explicitly configured with information that defines paths for directing traffic to all reachable networks. The router stores this information in a routing table, which it uses to make packet-forwarding decisions. Routers drop all local MAC-level broadcast traffic by default, thereby contributing to their effectiveness at isolating network traffic.

Summary

The two most common devices you are going to see in networks today are routers and switches.

Hubs Versus Switches



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Hubs Versus Switches

This slide shows the difference in how a hub and a switch work.

New Breed of Switches

- Embedded switches:
 - Routing capability
 - Perform load balancing based on the type of traffic requested
- Power over Ethernet:
 - Allowing VOIP devices to connect without external power

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

New Breed of Switches

Introduction

While new technology is constantly being developed, organizations are always looking for ways to reduce cost and the complexity of their existing networks. Convergence is a key trend that will continue. Convergence occurs when disparate technologies are merged into a single device. An area where this is occurring is with network-level devices. In the past, an organization might have had four different components in place. These can now be replaced with a layer 7 switch. The important thing to remember is that although convergence has benefits, there are also limitations.

Layer 3-7 Switches

Vendors have started to blur the functionality of traditional routers and switches and other devices, describing them as a Layer 3 switch or a Layer 4 switch. These blended devices use the Layer 3/4 indicator to refer to the level of the OSI model that the switch is interpreting for the processing of traffic. For example, a Layer 3 switch operates much like a router by using traditional switch techniques to process packets based on Layer 3 characteristics (IP addresses). A Layer 4 switch is similar, except that it processes traffic by inspecting Layer 4 characteristics, such as TCP and UDP port numbers. Layer 7 switches operate at the Application Layer and are able to evaluate application-specific information. For example, a web switch can process based on the type of web traffic requested.

Benefits

Blended devices offer improved performance to users over their traditional counterparts. For example, traditional routers using software for processing packets have evolved and are capable of processing over a million packets per second. A Layer 3 switch using hardware acceleration to inspect packets, however, can often process several million packets per second while offering the same features of a traditional Layer 2 switch on the same hardware.

Limitations

Although performance may be attractive to organizations wanting to reduce latency on the network with fast packet processing, blended devices do not typically offer the same flexibility and feature sets as their traditional

counterparts. For example, both a router and a Layer 3 switch may be capable of maintaining dynamic routing tables; however, the Layer 3 switch may be limited to the OSPF routing protocols. The traditional router has a much more varied selection of available routing protocols.

Challenges

The challenge to the consumer, however, is that not all blended devices are created equal. Although two vendors may both claim to produce a Layer 3 switch, the feature sets may vary widely between the two products. Some vendors may not offer the rate-limiting or traffic-shaping features found on traditional routers; other vendors may include this functionality in their product line.

Layer 4 and 7 Switches

Layer 4 switches can evaluate the socket, or connection information, which includes the TCP/IP port numbers. As such, they are able to perform some load-balancing to send different services to the appropriate servers (FTP, web, e-mail, and so on). However, they are not able to evaluate specific types of traffic within a given service. This is the benefit of Layer 7 switches. They are able to differentiate types of requests in a given service. For example, a web switch can evaluate the type of HTTP traffic and send each request to the most appropriate server. Streaming media requests, search requests, and database requests can be sent in the proper direction.

Also, in order to support VoIP, some switches can also provide electric power of the cabling to power devices like phones. This allows organizations to plug in network devices that require low grade power, without requiring an external power supply.

Summary

Before investing in blended devices, investigate the supported functionality on the device to ensure it meets the requirements of your organization. Don't assume that one vendor's implementation of a layer switch will match the functionality in a similarly named product. Prior to purchasing any new equipment, it is critical that an organization list its requirements to ensure the new device has the proper functionality.

Virtual LAN (VLAN) and Network Access Control (NAC)

- Virtual LAN (VLAN):
 - Allows segmentation of a switch into different networks, regardless of where a system is plugged in
 - Creates separate networks through software, not hardware
- Network Access Control (NAC):
 - Dynamic VLAN allocation
 - Isolates systems when they initially connect to the network
 - Enables systems to be scanned and checked prior to being put on a trusted segment
- 802.1x
 - Network-level authentication

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VLAN and NAC

Networks continue to be an avenue that malicious code uses to not only attack systems but spread across a network. While there are many pieces of software that can be installed on a host to protect them, hardware solutions are still one of the best ways to prevent an attack. Two of the most common methods of doing this is to limit the scope of a system through virtual LANs (VLAN) and preventing systems from connecting to trusted networks through Network Access Control (NAC).

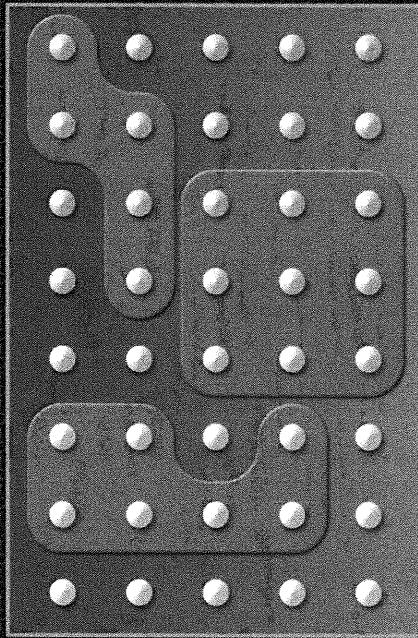
A virtual LAN (VLAN) allows you to take a large physical switch and regardless of where systems are plugged in, segment them into different networks based on function or access required. For example, if 100 employees and servers are all plugged into the same switch, you can limit the access or visibility by placing them on separate segments. Least privilege states that you give someone the least access he needs to do his job. For example, you can take all of the accounting employees and their respective servers and put them on a separate VLAN.

If 1,000 systems are plugged into a switch and the systems are on a flat network, when one system gets compromised, all **999** of other systems can also be compromised. However, if you create VLANs and put 100 systems on 10 separate VLANs and control traffic between the VLANs, when 1 system gets compromised, it compromises only 100 systems, not 1,000. VLANs help control the visibility of systems on a network.

One of the other problems today is that a laptop is out of the office for two weeks while an employee is traveling. This means the laptop is plugged into many untrusted networks and has a high chance of getting infected with malicious code. The infected laptop gets plugged back into the trusted network and infects several hosts. Network Access Control (NAC) allows systems to be placed on isolated VLANs until they have been scanned and properly patched, limiting their exposure to infecting other systems.

A key motto of security is prevention is ideal, but detection is a must. There is no way to completely prevent an attack from occurring. However if a single system becomes infected, you can prevent it from connecting to critical networks through NAC and stop it from spreading to a large number of hosts through VLANs. NAC and VLANs both have value by themselves, but together, they provide a robust measure of protection for networks.

Virtual LANS (VLANs)



Virtual LANS (VLANs)

This slide depicts how VLANs work within a switch.

Network Design

The student will be able to design basic network architectures using best practices.

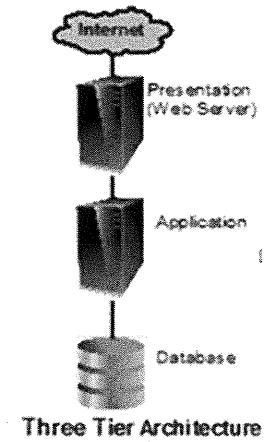
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Network Design

This section intentionally left blank.

Network Design Objectives

- Provide appropriate access from the internal network to the Internet
- Protect the internal network from external attacks
- Provide defense-in-depth through a tiered architecture
- Control the flow of information between systems



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Network Design Objectives

We would like to walk you through the fundamental steps of designing a basic network architecture, building upon what you have learned so far. In this example, one of the requirements for the network that we need to design is to allow internal users to access the Internet. Additionally, certain systems located on the company's network need to be reachable from the Internet, including:

- A Web server that displays information about the company and its products
- A mail server that allows the company's employees to send and receive e-mail
- A DNS server that hosts records for the company's public domain (such as "example.com")

According to these requirements, we need to provide limited access from the Internet to our network. However, aside from the servers listed previously, we do not want any Internet user to access our internal systems. Because of the link to the Internet, our defenses need to be designed to protect the network from external attacks.

Most of our design decisions will be based on the approach of "defense-in-depth," which advocates the use of multiple layers of protection to guard against failure of a single security component. One of the elements of defense-in-depth is the principle of resource separation, which we will use when dividing the internal network into several sections.

Network Sections (1 of 2)

- Public: Internet
- Semi-public (DMZ): Web, Mail, and DNS servers
- Middleware: Separate DMZ from the private network
- Private: Internal systems
- Locate firewalls:
 - Between the Internet and the other networks
 - Between the semi-public and private network
 - Between sections of varying trust levels

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Network Sections

Introduction

Security is always a balance between functionality and security. The key rule we always follow is to give an entity the least access they need, while still allowing them to perform their job. With network architecture, the key is to provide proper segmentation so that a person can access the appropriate data by reducing the risk of potential compromise.

Network Segments

If you look at the requirements for systems that will reside on our network, you will probably notice that they can be grouped into several categories, according to the type of information that they contain:

- Public: These resources reside on the Internet and, from the perspective of our company's network, cannot be trusted.
- Semi-public: These resources are our contributions to the Internet, and they take the form of Web publications, e-mail messages, and DNS records. Semi-public servers must be reachable from the Internet and might also have to access the Internet.
- Private: These are the company's internal systems, which are interested only in receiving resources from the Internet. We have no desire to provide any services from this category to users on the Internet. Therefore, we want to actively protect information that resides here.

Systems in each category serve a similar purpose and have common security requirements. This allows us to group resources within a category by placing them into a common network section. In such a design, our view of the networked world will be split into three sections: public, semi-public, and private. You can further subdivide based on potential risk and system functionality.

Network Sections (2 of 2)

- Three goals of network design:
 - Any system visible from the Internet must reside on the DMZ and cannot contain sensitive information
 - Any system with sensitive information must reside on the private network and not be visible from the Internet
 - The only way a DMZ system can communicate with a private network system is through a proxy on the middleware tier

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

This is resource separation at work; we placed systems with different security requirements into separate areas. In our example, grouping similar resources together will allow us to control how they interact with each other. Enforcement of such access restrictions is frequently the job of a firewall.

Firewall Placement

Installing a firewall is one of the most fundamental ways of protecting our systems from an external attack. The decision to use a firewall is simple enough, but a more interesting question is, "Where should we place it?" The idea is to position the firewall in a location that would allow it to control access or restrict traffic that crosses boundaries of network sections.

First, the firewall needs to be located in a place that allows it to ensure that any outbound traffic is legitimate. By the same token, we also want it to control inbound connections. This means that the firewall must be in a position that allows it to grant connection requests to our Web, mail, and DNS servers. The firewall should also be able to block inbound connections to systems on the private section of the network.

Network Traffic Flow

To help determine the optimal place for the firewall, consider the basic paths that traffic can traverse on our network:

- From private systems to the Internet
- From private systems to semi-public servers
- From semi-public servers to the Internet
- From the Internet to semi-public servers

Knowing the basic network paths really helps determine the best place for the firewall. It should be located at the intersections of the paths that we outlined previously. Now that the firewall's placement has been determined, let's look into adding additional defensive layers to further protect the network.

Providing Defense-in-Depth

Defense-in-depth is fundamental to the design of a secure network. It stems from the idea that software can have flaws, people can make configuration mistakes, and hardware devices can fail. To compensate for events like these, we do not want to rely on a single mechanism to defend our resources. Instead, we deploy multiple layers of protection to account for the possibility that one of them may fail.

In the context of our example, separating systems into several network sections is one defense layer.

Configuring the firewall to restrict how traffic crosses section boundaries is another. Yet another defense mechanism that we can employ is a device that operates in conjunction with the firewall when filtering traffic that leaves and enters the network.

Border Router

This device can take the form of a border router, placed between our Internet Service Provider (ISP) and our firewall.

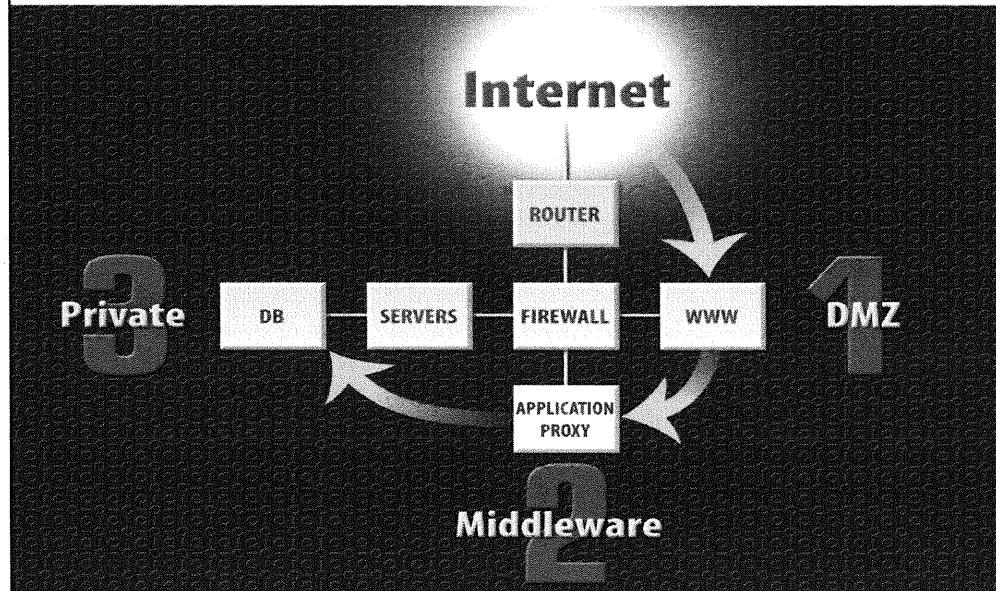
A border router can be used to filter out certain types of network traffic that obviously are unwanted. For instance, the router can be configured to block packets that claim to have come from invalid IP addresses, such as those allocated for private addresses used by RFC 1918. Similarly, only packets with source IP addresses that fall into the range assigned to us by the ISP should be leaving our company's network. Although a firewall can be set up to enforce restrictions like this, the router helps protect the firewall host itself from an attack. It also assists the firewall by taking some of the burden off it and leaving the firewall to process traffic that it is optimized to handle.

Another layer of defense can focus on individual sections of the company's network. On the off chance that one of our systems is compromised, we want to minimize the consequences as much as possible. One way to do this is to use switches instead of hubs on our network. This helps protect from an attacker setting up a sniffer on the compromised system and gathering information that can be used to attack other servers.

Summary

The concept of defense-in-depth can be applied to different types of networks, at the office as well as at home. For example, if you were putting together a home network, you might use a basic firewall built into your cable modem/DSL router to guard against direct attacks from the Internet. However, the firewall cannot protect you against all threats. For instance, you may receive a virus via an infected document that was e-mailed to you or handed to you on a floppy. It would be a good idea to install anti-virus software on your workstations to account for such attack vectors. A malicious Web site that you visited might attempt to exploit a vulnerability in your Web browser. A firewall and anti-virus software might help combat this threat, but keeping up with application and OS patches provides another highly effective layer of protection. Similarly, a malicious worm might require access through some service port and be blocked from your system by the firewall. Although some exploit might penetrate a firewall, a securely configured authentication system or intrusion detection system can thwart the attack. Not relying on a single security mechanism to protect you against attacks is the fundamental principle of defense-in-depth.

The Final Design



The Final Design

This slide presents our final design. Traffic passes to and from the Internet through the border router, which filters out traffic that is obviously not wanted using access control lists. The firewall controls the flow of traffic to and from semi-public (DMZ) and private network sections. Switches within each section guard against sniffers that might be installed on compromised systems. To further enhance security through the principles of defense-in-depth, all hosts (servers and workstations) in our overall configuration would actively run anti-virus software with the latest signature updates. Hosts might also have host-based firewalls applied or host-based intrusion detection systems.

Summary

- Understanding network technologies, cabling, components, physical and logical topologies, network devices, and upper-level switches is vital to creating and maintaining a secure network
- To secure a network, we must understand how it works
- Security must be embedded into the network and not be an afterthought

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

Introduction

Let's summarize the critical components of network architecture and design.

Only by understanding and knowing how an architecture is connected and configured can an organization implement proper security. From how the wires are connected to the data that flows over those wires, knowledge is power in making sure a network is properly secured.

Physical Topology

A physical topology describes how cables and devices are wired together to form a functional network. Among physical topologies, star is most commonly used on modern LANs. It is the most versatile because it can support all of the media access protocols that we have examined: Token Ring, FDDI, ATM, and Ethernet. Network nodes connected according to a physical star topology are wired to a common device, such as a hub or a switch.

Logical Topology

A logical topology is independent of the physical topology and describes low-level communication mechanisms that allow systems on the network to exchange signals with one another. Ethernet is the most popular LAN protocol for sending signals over the wire. On an Ethernet network, only a single frame can be transmitted at a time to prevent collisions. Ethernet nodes are required to monitor the status of the signals that they issue to detect collisions and resend the frames if necessary. Systems on Ethernet networks are identified by unique MAC addresses, which usually are embedded into their network cards.

Summary

Too often organizations buy the state-of-the-art components and are surprised when a security breach occurs. Components do not make a network secure. Only by understanding how components on a network work and through a proper network architecture design can an organization achieve a secure network.

Module 3: Protocol Stacks and IP Concepts

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 3: Protocol Stacks and IP Concepts

This section intentionally left blank.

Protocol Stacks and IP Concepts

SANS Security Essentials I: Networking Concepts

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Introduction – Protocol Stacks and IP Concepts

You cannot get very far in any subject without a good basic understanding of that topic's fundamentals. You are studying information security, and because much information that needs to be secure is transmitted across networks, a good understanding of how these networks actually work is critical. A solid understanding of the inter-workings of networks allows you to be more effective in recognizing, analyzing, and responding to the latest (perhaps unpublished) attacks. This chapter is an introduction to the core areas of computer networks and protocols.

Objectives

- Network protocols
- IPv4
- Network addressing
- Domain Name System (DNS)
- IPv6

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

We begin with a quick refresher on protocols and protocol stacks, the basic building blocks of network communications. We examine and compare two of the most common examples—the OSI reference model and TCP/IP.

After you have mastered the basics of protocols, we show you how they work together to structure network transmissions into frames and packets as they are sent across the wire. Additionally, we examine the Internet Protocol (IP) packet header to see what you can learn from it.

We continue by discussing network addressing and host naming. That is, how computers tell the network which remote computer they would like to communicate with and how humans distinguish one computer from another.

Once you have an understanding of how IP networks work, we look at IP version 6 (IPv6), which was designed to address some of the limitations present in IPv4 networks. We discuss how IPv6 varies from IPv4, how IPv6 is addressed, and some of the features of IPv6. We go on to examine the IPv6 header and finish up with a brief discussion of the major differences and advantages of IPv6 over IPv4.

We round out the module with a discussion of the Domain Name System (DNS) and how it makes networking more user-friendly by providing a mechanism to convert IP addresses to names.

Network Protocol

The student will understand the properties and functions of network protocols and the network protocol stacks.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Network Protocol

This section intentionally left blank.

What is a Network Protocol?

- A network protocol is an agreement or rules of engagement for how computer networks will communicate
- Entities exchanging messages are the network's software and hardware
- Protocols define the format and order of messages and the actions to be taken upon receipt of the messages
- Protocol stacks are a set of network protocol layers that work together to implement communications

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

What is a Network Protocol?

Introduction

In the broadest sense, a protocol is nothing more than an agreement of how different entities will act and react in certain circumstances. A medical protocol prescribes a course of treatment for a certain disease. A diplomatic protocol is the basis for a formal treaty that, for example, might specify how two nations will allow free trade along a common border.

Network Protocols

A communications/network protocol establishes an agreement between network entities, such as hosts and servers, for how they will communicate. When protocols are worked out in advance, they are effective and efficient. If any participant breaks the protocol, the communication gets confused or can break down all together. You have probably been in a situation in which you had "interoperable" hardware or software products that were based on the same standards but were not actually compatible. Odds are, one or both of those products deviated from the standard and implemented the protocol differently. This is why we require strict conformance to standard protocols.

What Are the Purposes of Network Protocols?

There are three basic purposes for communications protocols:

- To standardize the format of a communication
- To specify the order or timing of communication
- To allow all parties to determine the meaning of a communication

As long as both sides of the communication are using the same protocol and implement it properly, communication will be successful.

Protocol Stacks

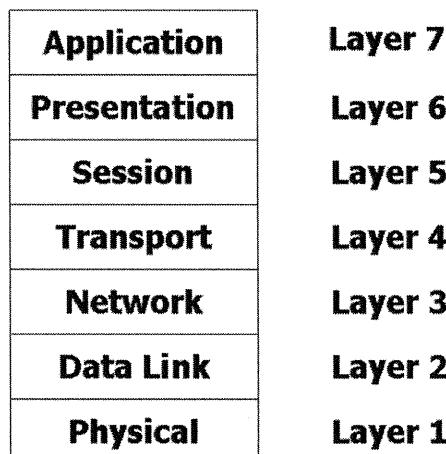
If two computers want to communicate, they need to follow a specific set of protocols for communications to succeed. There are a number of protocols involved. Some protocols concern themselves with breaking up a transmission into

smaller bunches of data called packets. Some make sure that each packet has the proper information in the proper locations. Others describe how information is copied from your computer to the network cable. Still others ensure that packets all get to the right place in the proper order. Even with a transaction as seemingly simple as fetching a Web page, a number of protocols are required to allow the communication to succeed. In computer communications, these layered protocols are referred to as a protocol stack.

Summary

In order for two or more entities to be able to communicate, they need to have standard rules of engagement. A protocol provides those rules so computers anywhere in the world can communicate. In order to make the protocol easier to manage, it is broken down into a protocol stack where each layer receives a service from the layer below it and provides a service to the layer above it.

The OSI Protocol Stack



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

The OSI Protocol Stack

Introduction

In order to be able to allow computers to communicate across a network, a protocol stack is required. The OSI model is one of two protocol stacks that we are going to cover. The OSI model is used to describe and talk about the various layers in a protocol stack.

OSI Model

The standard reference model for protocol stacks is the International Standards Organization's (ISO) Open Systems Interconnect (OSI) model. The OSI model divides network communications into seven layers.

Physical Layer

The Physical Layer handles transmission across the physical media. This includes such things as electrical pulses on wires, light pulses on fiber, connection specifications between the interface hardware and the network cable, and voltage regulation.

Data Link Layer

The Data Link Layer connects the physical part of the network (cables and electrical signals) with the abstract part (packets and data streams).

Network Layer

The Network Layer handles the network address scheme and connectivity of multiple network segments. It describes how systems on different network segments find and communicate with each other.

Transport Layer

The Transport Layer interacts with your data and prepares it to be transmitted across the network. It is this layer that ensures reliable connectivity from end-to-end. The Transport Layer also handles the sequencing of packets in a transmission.

Session Layer

The Session Layer handles the establishment and maintenance of connections between systems. It negotiates the connection, sets it up, maintains it, and makes sure that information exchanged across the connection is in sync on both sides.

Presentation Layer

The Presentation Layer makes sure that the data sent from one side of the connection is received in a format that is useful to the other side. For example, if the sender compresses the data prior to transmission, the Presentation Layer on the receiving end would have to decompress it before the receiver could use it.

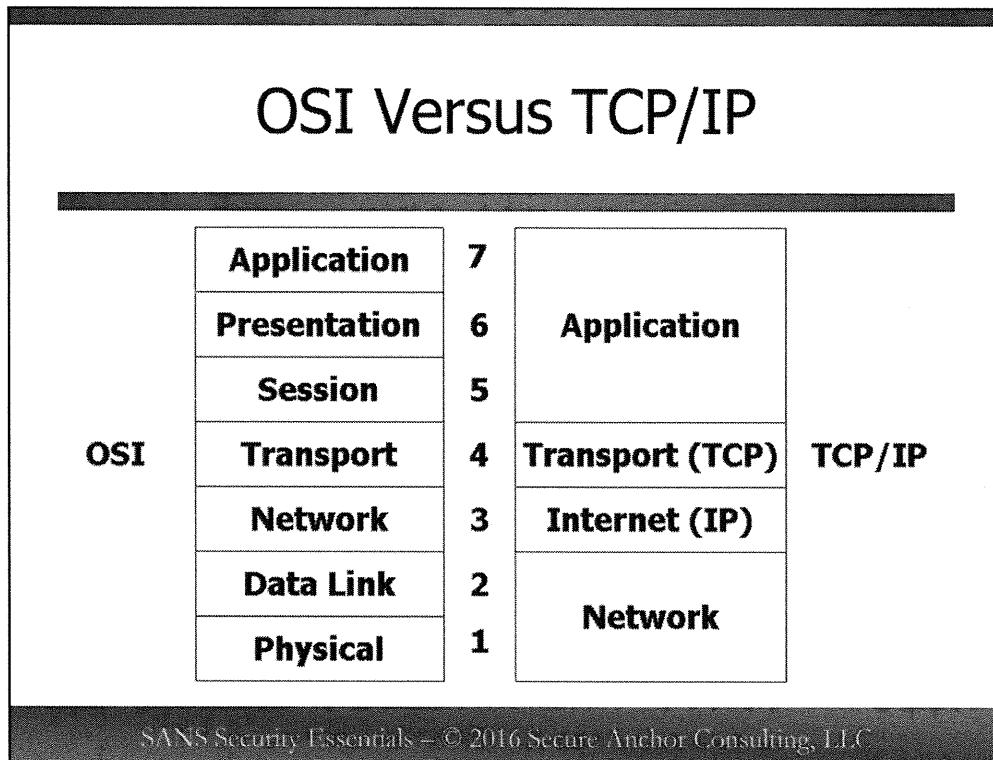
Application Layer

The Application Layer interacts with the application to determine which network services will be required. When a program requires access to the network, the Application Layer will manage requests from the program to the other layers down the stack.

Summary

The OSI model is a reference model. It is followed when building network applications; however, in many cases, some of the layers are combined together. Still, the core functionality is present. Most protocol stacks do not have all seven layers clearly delineated. However, understanding the OSI model is important because the model serves as a common point of reference and a kind of verbal shorthand. Network engineers and vendors talk about "Layer 2 switches" or "Layer 3 protocols." Because the layers to which they are referring are the OSI model layers, understanding what each layer does will go a long way toward understanding the conversation and securing your network services.

OSI Versus TCP/IP



OSI vs. TCP/IP

Introduction

Many people ask which protocol stack is better, OSI or TCP/IP. The answer is both. In practice, both are utilized and need to be understood by the student. When talking about the protocols and referencing the layer a protocol operates at, OSI is utilized. For example, we always say that routing is a layer 3 function. However, when we actually implement the protocols, we use the layered functionality of the TCP/IP model.

TCP/IP Model

The TCP/IP stack has only four layers: the Network Layer, the Internet Layer, the Transport Layer, and the Application Layer. Even though the stack has only four layers as compared to the seven-layer OSI model, it still performs the same functions. It just means that because there are fewer layers, each layer has to do a little more work.

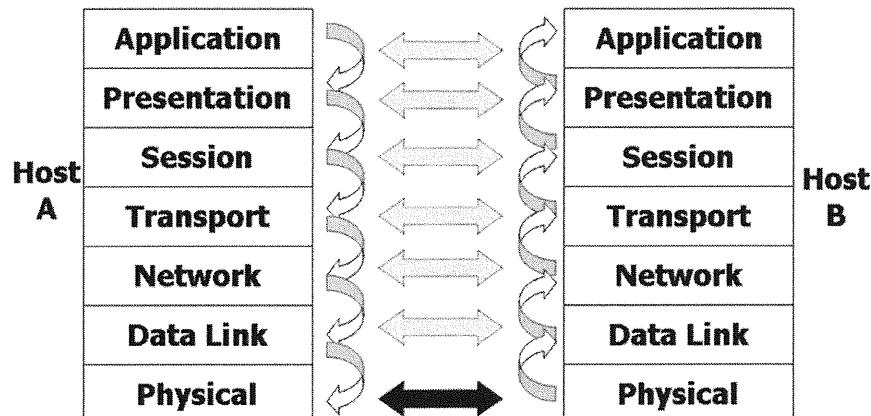
Comparison

As you can see, the OSI model is more granular. The OSI model splits apart some functionality that was combined in the TCP/IP model. The Network Layer in the TCP/IP model comprises both the Physical and the Data Link Layers in the OSI model, while the Application Layer in TCP/IP encompasses the Application, Presentation, and Session Layers of OSI. The OSI model is more detailed because it was designed to support protocols other than just TCP/IP. By creating more layers, the designers made it easier to break down the functionality of each protocol and build more specific interfaces and linkages between the layers.

Summary

Even though each model breaks down the functionality a bit differently, you should realize that no matter which model you use, it must perform all the functions required to take a piece of application data, place it into a packet, put that packet on the wire, and deliver it safely and efficiently to its destination.

How Protocol Stacks Communicate



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

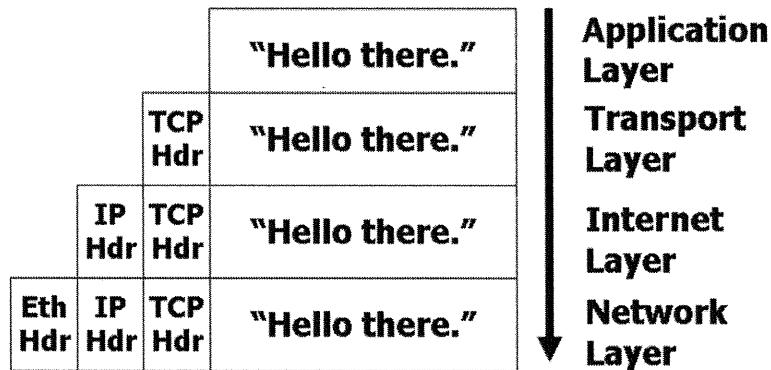
How Protocol Stacks Communicate

The basic principle of stack-based communication is that data from one layer of the stack can be understood only by the corresponding layer on the remote computer. In other words the Application Layer on Host A exchanges information with the Application Layer on Host B, and the Transport Layer on Host A exchanges information with the Transport Layer on Host B, and so on. However, each layer is only aware of the requirements that must be fulfilled to communicate with the corresponding layer on the remote host. Each layer is not aware of the contents of the data from other layers and, in fact, does not need to be aware of specifics of the other layers for communications to succeed. This layer independence does have security implications.

For example, assume you want to pretend to be another computer and spoof its IP address, perhaps to give an incorrect DNS answer. You would have to be in the communication path so you could sniff the packet and keep it from reaching the real DNS server. Then you could craft the answer. But what if you did not carefully spoof the IP header? Then the TTL would be wrong. If the IP Layer and UDP Layer work together, they could probably notice a change. But they don't. This is an important security principle. Layer independence makes it easier and faster to write and maintain networking software, but your security systems need to consider all of the information in all of the layers. What this means is that each layer needs to deal only with its own communication requirements and then pass the data down the stack so each subsequent layer can satisfy its own requirements. Each layer takes the data from the layer above it, satisfies its own requirements by adding its own data, and then passes it to the next layer down the stack. On the receiving end, the data is passed up the stack with each subsequent layer removing the data added by its peer layer and passing the data up the stack until, finally, the Application Layer receives the data.

How TCP/IP Packets are Generated

"Hello there."



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

How TCP/IP Packets are Generated

Introduction

It is important to understand how a packet is generated as it moves through the TCP/IP stack. Ultimately, each layer on the sender needs to communicate with the same layer on the receiving computer. However, they cannot directly talk because you must go down the stack, across the network, and back up the stack on the receiving system. The way this is accomplished is by having each layer add a header as you go down the stack on the sender and each system remove a header on the receiving system as it goes up the stack. By performing this function, the header that is created by a given layer on the sender is received by the corresponding layer on the receiving system.

Application Layer

To start with, the Application Layer takes information from the application itself. In this case, we send the phrase "Hello There" to another computer. A program gives the "Hello There" to the stack's Application Layer, which creates an empty packet and places the "Hello There" inside. The Application Layer then sends the packet to the Transport Layer.

Transport Layer

The Transport Layer takes the packet and adds a header to it. The header has all the information that the Transport Layer on the other side of the connection needs to determine what to do with the packet. After the transport header is put on the packet, it is given to the Internet Layer.

Internet Layer

The Internet Layer puts another header in front of the packet. Like the Transport Layer before it, this header gives information for the Internet Layer on the other end. After this header is attached, the packet is sent to the Network Layer.

As you probably have guessed by now, the Network Layer will put its own header on the packet. This header will assist the routers and gateways between the two machines in sending the packet along its way. After this final header is placed on the packet, it is put on the wire and sent to its final destination.

Receiving System

What happens when the remote computer receives the data? The operation starts again, but this time in reverse. The Network Layer begins by stripping off the header its counterpart put on the packet in the first place and then passes the rest of the packet up to the Internet Layer. The process continues up through the rest of the stack, each layer removing only the information placed in the packet by its counterpart in the sending host's stack until the original "Hello There" string reaches the remote application. This process of stripping off one layer's headers and passing the rest of the packet up to the next higher layer is known as decapsulation.

Summary

Each layer in the stack adds its own header to the packet and passes it along to the next lower layer. It encapsulates the packet it was given in protocol headers of its own before passing it on to the next lower layer, which performs its own encapsulation. The process is performed in reverse on the receiving system.

IP (Internet Protocol)

- Works at the Internet Layer of the TCP/IP stack
- The core routing protocol of the Internet
- Deals with transmission of packets between end points
- Defines the addressing scheme for the Internet

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IP (Internet Protocol)

IP is the basis for all communication on the Internet. It's so important that it even gets its own layer in the TCP/IP stack! The primary purpose of IP is to handle the transmission of packets between network end points, usually single hosts identified with a unique address. IP includes some features that provide basic measures of fault-tolerance (time to live, checksum), traffic prioritization (type of service), and support for the fragmentation of large packets into multiple smaller packets (ID field, fragment offset).

IP is singularly focused on getting packets from point A to point B on the network as quickly and efficiently as possible. IP does not provide any mechanisms for guaranteed delivery or delivery in sequence. Instead, it relies on upper-layer protocols and applications to provide those mechanisms, as appropriate, for the application.

IP defines the IP addressing scheme that allows each host to be uniquely identified. It also defines the rules used to route packets between hosts, whether close or separated by large distances.

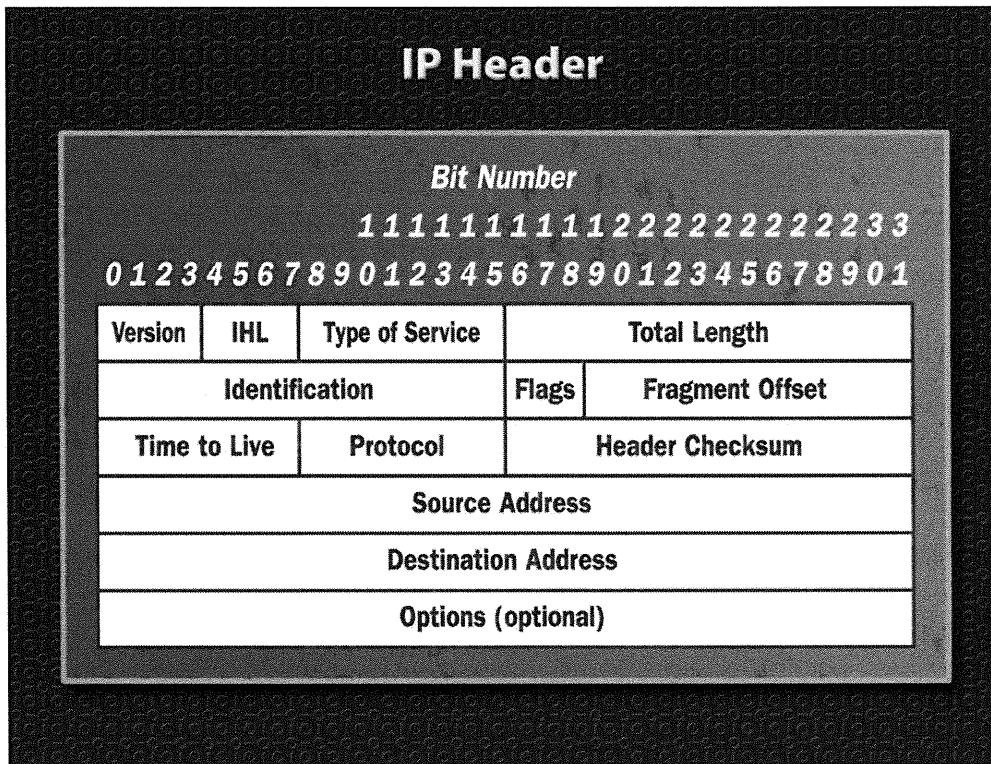
IP Packets

The student will have a fundamental understanding of how the IP protocol works.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IP Packets

This section intentionally left blank.



IPv4 Header

Here you see a diagram of how the bits inside an IP packet header are laid out. Pay particular attention to the way the diagram is labeled, as this is the standard way of looking at a packet header. Across the top, the bits are numbered from 0 on the left to 31 on the far right, for a total of 32 bits. 32 bits equal 4 bytes. When counting within packet headers, always start counting bits and bytes starting from 0.

Most IP headers will have no options set and will have a length of 20 bytes. The first byte is byte 0, and if no options are set, the last is byte 19.

If options are used, the header will be longer than 20 bytes. The options field can be of variable length, but must end on a 4-byte boundary.

Some IP options are:

- Record Route: Tells a router to add its IP address to the options field.
- IP Timestamp: Tells a router to write a timestamp into the options field.
- Strict Source Routing: Allows the sender to specify the exact route a packet should take to the destination.
- Loose Source Routing: Allows the sender to specify a list of routers a packet must pass through. It may also traverse other routers if required.

Although options do have valid uses, especially for network troubleshooting, they are rarely used by legitimate traffic.

IPv4 Header Key Fields

All the IP header fields are important. If a packet is missing even one, it probably will not reach its destination; or if it did, it would most likely be unusable. For our purposes, though, some fields are more important than others. Let's take a look at some of them.

As we go through these fields, keep in mind that they are part of the IP header, which resides at the Internet Layer. These fields are common to all IP packets, but the individual protocols that depend on IP—TCP, UDP, and ICMP—each has its own headers, which we examine later.

IP Version - 4 Bits

This contains a short integer corresponding to the Internet Protocol version used to create this packet. The most common value is 4 for IPv4. IPv6 is becoming increasingly popular and will use a value of 6.

Protocol - 8 Bits

This is another integer that denotes the exact type of IP message encapsulated in this packet. Although the meanings of the possible values are standardized, the values themselves are fairly arbitrary. For a TCP packet, expect this value to be 6. For a UDP packet, the value is 17 decimals. ICMP packets carry a value of 1 in this field.

Time To Live (TTL) - 8 Bits

A packet's TTL specifies how many hops a packet is allowed to take before it reaches its destination. For example, a typical TTL value of 32 says that the packet can go through a maximum of 32 routers on its way to the destination. Each time the packet passes through a router, the router subtracts one from the TTL and places this new TTL in the packet when it sends it on its way. If the new value is zero, however, instead of forwarding the packet, the router drops the packet. If the router's administrator is friendly, it sends an ICMP "Destination Unreachable" packet back to the original sender to let it know that its packet was never delivered.

TTLs guard against routing loops, where two or more poorly configured routers repeatedly exchange the same packet over and over again in the mistaken hope of getting it to its final destination. The packets can keep looping around and around indefinitely, and as more packets arrive, they can be added to the loop. Pretty soon, the routers involved are able to do nothing more than continually try to deliver the same undeliverable packets and they are unable to accept any new traffic, bringing down that part of the network. By using TTLs, packets in a routing loop are guaranteed to expire quickly and stop competing for network resources.

Fragmentation - 16 Bits (13 Bits Fragment Offset and 3 Bits for Flags)

Sometimes a router encounters a packet that is too big for it to retransmit all at once. Rather than signaling an error to the sender, most routers simply fragment the packet or break it up into two or more smaller individual packets, and then send them on their way. This is quite common when packets traverse different types of network links on their way across the Internet because some networking technologies have different maximum packet sizes. Packets can usually be split at any point. The fragment offset field tells the sender where this particular fragment falls in relation to the other fragments of the original larger packet.

Source Address and Destination Address - 32 Bits Each

Both computers involved in the communication, and all the routers and network devices between them, need to know who is talking to whom in this packet. The source address contains the IP address of the packet's sender. The destination address lists the IP address of its intended recipient.

Fragmentation Attacks

One might see several interesting attacks while examining the fragment offset fields of malicious packets. For example, consider an intrusion detection system that is set to sound the alarm if it sees the string "rm -rf /" (a Unix command to delete all files on the system). Instead of sending a simple string like this, perhaps an attacker will fragment it into two separate packets, one containing "rm -" and the next "rf /." Each string individually would be relatively harmless. It is only when the receiving computer's TCP stack reassembles the fragments back into a single packet that the strings reveal their hidden evil intention.

Another interesting attack is to send several packets that are supposed to be fragments of a single packet but actually contain overlapping or contradictory fragmentation offsets. The first packet might claim to contain bytes 0 through 237 of a packet, while the next claims to contain 150 through 400. Not only would these two payloads overlap each

other if reassembled into a single packet, but the second packet might also contain different bits in the supposedly overlapping area. Some IP stacks (including some fairly new ones) cannot handle this ambiguity. This was a popular method for attackers to crash remote IP stacks and force their targets to drop off the network.

Another popular method is to send thousands of initial fragments, but never send the rest of the packets. The IP stack on the target has to keep these in a buffer so it can reassemble the entire packet after it receives all the pieces. If the other pieces never come, the buffer can fill up and cause the IP stack to crash or become unresponsive until some of these fragments time out and are thrown out.

In the last few years, most IP stacks have incorporated better countermeasures against these sorts of attacks. Still, IP fragmentation problems continue to crop up fairly often, so you should be on guard against them.

Network Addressing

The student will understand
the essentials of IP addressing,
subnets, CIDR, and netmasks.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Network Addressing

This section intentionally left blank.

Addressing Basics

- Addressing provides a mechanism to identify endpoints in a communication
- Each node on an IP network has a unique IP address
- A portion of the address denotes the network and the remainder represents the host

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Addressing Basics

For a second, let's think about your old-fashioned snail mail mailing address. It provides a unique way of identifying your house. A network address is no different. An IP address just uniquely identifies a node in a network. The address permits nodes in the network to identify each other for the purpose of communicating. Part of the IP address is used to identify the network the node is on. The remainder represents the host.

To extend the analogy of your mailing address, consider how the Postal Service delivers mail to your house. Your ZIP code tells the Postal Service which neighborhood to deliver the mail to, and your house number and street address tell the letter carrier which specific house to deliver the mail to.

An IP address is the same. The network portion of the address permits the routing of the packets to the correct local area network, and the host portion of the address tells the LAN which machine should receive the packet.

Two Parts of an Address

- An address is broken down into network and host portions
- All nodes on the same network segment must have the same network address, but different host addresses
- Two nodes can have the same host address only if they are on different network segments

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Two Parts of an Address

In order for the network communication to succeed, a couple of things must be true about the IP address. First, all the nodes on one network segment must have the same network address. Think about it. What if your neighbor's house had a different ZIP code than your house? The Postal Service's routing of your mail would become much more complicated and confusing, resulting in an increased risk of mistakes being made in the mail delivery. In order for the delivery to be efficient, your neighbors need to have the same (or similar) ZIP code to you. This is true of networking as well. Your neighbor and other nodes in your immediate vicinity must have the same IP address.

Second, two hosts on the same network must not have the same host address. What would happen if your house and your neighbor's house had the same house number? The letter carrier would not know which house to deliver the mail to! The same is true of network addresses. The host address needs to be unique within the network or the network will not know where to deliver the packets.

Of course someone living on the next street over can have the same house number as you, because they live on a different street, so the letter carrier knows the addresses are different and will successfully deliver to both houses or any other house on any other street with the same house number as yours. Networks are the same. Nodes on different network segments can have the same host address because they are on different networks the communication will succeed.

IPv4 Addresses and Subnets

- Each node on an IP network has a unique IP address
- IP addresses are denoted as four numbers separated by periods (dotted quad)
Example: 135.118.231.10
- Subnet mask (or netmask) defines which portion of the address is the network address and which portion is the host address
- Class A address - 1-127
 - N.H.H.H
 - 255.0.0.0
 - /8
- Class B address - 128-191
 - N.N.H.H
 - 255.255.0.0
 - /16
- Class C address - 192-223
 - N.N.N.H
 - 255.255.255.0
 - /24

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IPv4 Addresses and Subnets

IP addresses are represented using a dotted quad notation, four numbers separated by periods. All four of these numbers will be in the range of 0 to 255. Remember back to our discussion of binary numbers; 255 is the largest number that can be stored in 1 byte. Therefore, storage of an IPv4 address takes 4 bytes or 32 bits.

The Internet relies on many assumptions in order to function properly. One of these assumptions is that each organization connected to the Internet will use unique IP addresses for its computers. These days, when you connect to the Internet, your ISP assigns you a block of IP addresses to use for hosts at your site. If you are a home user, you typically get only one address (or maybe two or three if you have that many computers). Companies and other organizations usually get substantially more.

Subnet Masks

How can you tell if a given node is on your local network segment, or if the traffic should be forwarded through a router to a different network segment? The answer is the netmask (sometimes referred to as the subnet mask). The netmask determines which portion of the address identifies the network address and which portion identifies the host address.

Netmasks and CIDR

- Netmasks or subnet masks provide a method for identifying which portion of an address is the network and which portion is the host
 - The network portion of the netmask is all binary 1s
 - An IP address of 172.20.15.5 with a netmask of 255.255.0.0 is 16 bits of network and 16 bits of host
 - Net ID (172.20) and host ID (15.5)
 - Network is specified as 172.20.0.0
- CIDR provides a shorthand way of specifying which portion of the address is the network and which portion is the host
 - With an IP address of 172.20.15.5 with a netmask of 255.255.0.0, the network can be specified as 172.20.0.0/16
 - The /16 means that 16 bits are used to identify the network
- Some addresses are reserved for special use
 - The first address in a network range specifies the network address
 - The last address in a network range is reserved for the broadcast address

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Netmasks and CIDR

A subnet mask or netmask is a bitmask that shows where the network address ends and the host address begins. Recall that IPv4 addresses are 32 bits. Some portion of those bits are allocated to define the network, and the remainder defines the host portion of the address. If the server has an IPv4 address of 172.20.15.5 and 16 bits of the address are reserved for the network, then the network portion would be 172.20 and the remainder would be the host portion, 15.5.

Netmasks are expressed in a notation similar to IP addresses; however, in order to understand netmasks, we need to think in binary for a bit. In a netmask, any bit in the netmask that is interpreted as part of the network number is represented by a 1 in the netmask. Those that define the host portion are set to 0. For example, in binary, the netmask would be 11111111 11111111 00000000 00000000, or, converting that to decimal 255.255.0.0. But what good is the netmask? Earlier, we used the term bitmask. Remember, computers are good at performing arithmetic on binary numbers. It turns out the bitmask provides an easy way for a computer to distinguish the network portion from the host portion by doing a masking operation.

172.20.15.5	10101100 00010100 00001111 00000101
255.255.0.0	11111111 11111111 00000000 00000000

10101100 00010100 00000000 00000000

The network portion is 172.20.0.0.

Historical Description of IP Addresses - Classes

In the not too distant past, when you needed a block of IP addresses for your network, you told your service provider how many hosts you had or thought you were going to have, and the provider would allocate an entire network number for your use. These network numbers came in three fixed sizes, so the number of hosts you needed to connect determined the class you were assigned. This original system was called "classful" routing.

Class A networks are allocated in the range of 1.0.0.0 through 127.255.255.255. The first bit in the first byte of a Class A network always begins with "0" and the subnet mask is 255.0.0.0. With 8 bits for the network there can only be 126 Class A networks on the Internet (because 0 and 127 are reserved). However, with 24 bits reserved for hosts, each network can accommodate over 16,000,000 hosts.

Class B networks are allocated in the range of 128.0.0.0 through 191.255.255.255. The first two bits in the first byte of a Class B network always begin with "10" and the subnet mask is 255.255.0.0. With 14 bits for the network, there can be over 16,000 Class B networks on the Internet, and with 16 bits reserved for hosts, each network can accommodate up to 65,534 hosts.

Class C networks are allocated in the range of 192.0.0.0 through 223.255.255.255. The first three bits in the first byte of a Class C network always begins with "110" and has a subnet mask of 255.255.255.0. With 21 bits for the network, there can be over 2,000,000 Class C networks on the Internet. With 8 bits reserved for hosts, each network can accommodate up to 254 hosts.

Technically, two other classes exist: Class D and Class E. Class D networks are used for IP multicasting, a special form of network transmission that is intended for multiple hosts, often on different LANs. Class E networks were reserved for future expansion. Neither is in common use today, though many Unix systems configure multicast networking at boot time by default.

Although this method of allocating IP addresses worked when the Internet was young, in practice, it became rather inflexible as the Internet grew. Under classful routing if you contacted your network provider and asked for address space for a company with 25 network nodes, the smallest range the ISP could provide you was a class C network with 254 valid addresses, thus wasting the majority of the address spaces. This was inefficient and created a shortage of IP addresses as the Internet grew.

Although it is good to know about classful routing for historical purposes, classful routing has been supplanted by CIDR.

CIDR

CIDR (pronounced like cider) is an abbreviation for Classless Inter-Domain Routing. CIDR uses variable length subnet masks (VLSM) to allocate IP addresses to subnets according to individual needs. Thus, the network/host division can occur at any bit boundary in the address. The process can be recursive, with a portion of the address space being further divided into even smaller portions through the use of masks that cover more bits. Because the legacy network class distinctions are ignored, the new system was called classless routing.

Under CIDR, if you request addresses for 25 nodes, the ISP can break up the Class C network into smaller segments and provide you with an appropriately sized network, reducing waste. A 5-bit host range provides room for 32 addresses, adequate for the example. What would the subnet mask for this network be? Again, go back to our binary representation. In this example, 5 bits are used for hosts, leaving 27 bits for the network portion. In binary:

11111111 11111111 11111111 11100000

Or, converted to dotted-quad notation:

255.255.255.224

CIDR/VLSM network addresses are now used throughout the public Internet, although they are also used elsewhere, particularly in large private networks.

CIDR also provides us with shorthand for specifying which portion of the address is the network portion. Instead of specifying it in terms of a dotted quad, we are permitted to specify the number of bits in the network portion by appending the network specification with a "/" followed by the number of bits in the network portion.

Let's go back to our example of the 172.20.15.5 address in a Class B network (255.255.0.0 mask). As we determined earlier, the network portion is 172.20.0.0 with 16 bits for network and 16 bits for host. Using the shorthand, we can specify that network as 172.20.0.0/16.

This is a fairly obvious example, but the notation works just as well for less obvious outcomes. Let's say that when we requested a network block for the 25 addresses, the ISP gave us a portion of the Class C network starting from 172.20.50.0 with 27 bits for network and 5 bits for host. Using the shorthand notation, the network portion is 172.20.50.0/27. We also know that we have 5 bits for host, or a total of $2^5=32$ addresses. So we know that we have the addresses from 172.20.50.0 to 172.20.50.31.

Network Addresses and Broadcast Addresses

Up until now, we have been treating all the host addresses in the network the same. In reality, a couple of them are special. The lowest host address (all 0s) in a range is always reserved for the network address. The highest host address (all 1s) in a range is reserved for the directed broadcast address.

For example, if you have a network of 172.20.15.0/24, 172.20.15.0 is referred to as the network address and is reserved for historical reasons. 172.20.15.255 is referred to as the broadcast address.

Private Network Addressing

- Private address blocks (RFC 1918) are not routed on the Internet:
 - 10.0.0.0 -> 10.255.255.255
 - 172.16.0.0 -> 172.31.255.255
 - 192.168.0.0 -> 192.168.255.255
- Private Network Addresses make more efficient use of public IP addresses
- They make it difficult to trace information back to the source
- They increase security of internal machines
- They require NAT to access the Internet or other networks
- Loopback addresses used to identify an address local to the machine:
 - 127.0.0.0/8
 - Should never be seen on your network

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Private Network Addressing

Not every host capable of accessing the Internet has a direct connection. These days, computers are (or should be!) behind firewalls of some sort. They can also use Network Address Translation so that the IP addresses in use on the internal LAN are automatically mapped to a different IP address or set of IP addresses when they traverse the firewall and go out to the Internet. If no one on the Internet can ever see these addresses, why should an organization bother to request an address block from its ISP? Even more to the point, why should the ISP waste addresses by allocating them to a customer when these addresses never will be routed over the Internet?

It turns out that the answer to each of these questions is, "They do not have to." The Internet Assigned Numbers Authority (IANA), the ultimate authority for IP address assignments, has designated three sets of private address blocks that never can be routed over the Internet and, therefore, are free for anyone to use as they wish within their own networks. Because these addresses cannot traverse the Internet, it does not matter if two, five, or 10,000 different sites pick the same address to use on their internal networks. As long as the traffic is translated to routable IP addresses before it goes out onto the Internet, the actual internal network numbers used do not matter a bit.

Private Network Allocations

10.0.0.0/8

172.16.0.0/12 - 172.31.0.0/12

192.168.0.0/16

In addition to these private network numbers, there is another special class of non-routable IP addresses. These addresses fall into the range 127.0.0.0/8 and are referred to as loopback addresses. The loopback addresses are never routed over any network, not even a local LAN segment. They are to be used only as pseudo IP addresses that always refer to the local host. In other words, every host responds to traffic addressed to

127.0.0.0/8, but only if the packets came from the same local host. Loopback addresses are often used by services that must contact other services running on the same machine. They are convenient because no matter what the system administrator has set, the host's IP address to the services can always refer to the local host by using a loopback address, usually 127.0.0.1. Note, however, any time you see a 127.0.0.0/8 packet on the network, either something is improperly configured, or someone is trying to attack you.

Broadcast Addresses

- Directed broadcast:
 - Host bits of address set to all 1s
 - For network 172.20.15.0/24, the broadcast is 172.20.15.255
 - Will be sent to all hosts with a given network address
 - Passed on by routers
- Limited broadcast:
 - All address bits set to 1s (255.255.255.255)
 - Limited to local segment, not passed on by routers

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Broadcast Addresses

We mentioned before that you should never use host addresses of all 0s or all 1s. In a /24 subnet, these would be the addresses x.x.x.0 and x.x.x.255. That's because these are reserved for a special kind of network transmission known as a broadcast. A broadcast packet is a single packet that is processed by every IP stack on the LAN.

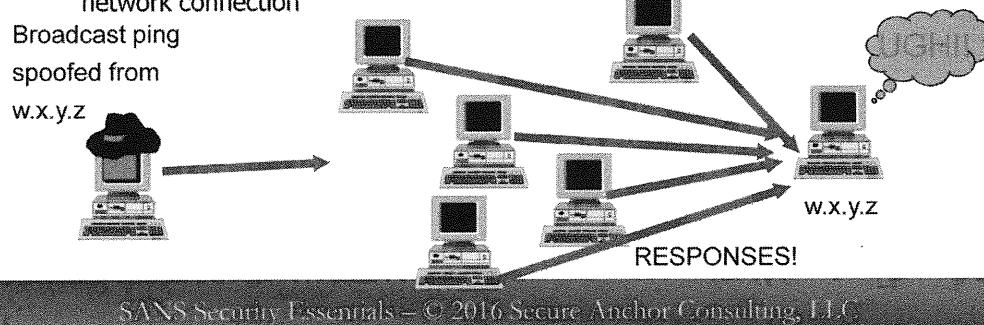
Types of Broadcast Packets

There are actually two types of broadcast packets that you might see. The first is called a net directed broadcast, which is a fancy way to say that the network number bits in the broadcast address are the same as those in the host's IP address. The host bits are still all 1s, though, to differentiate these packets from regular traffic. Net directed broadcasts are, as the name implies, intended for all hosts with a specific network number. Routers and gateways usually pass these along to other parts of the same network that might happen to reside on different physical segments of cable.

The second type of broadcast is referred to as a limited broadcast. Packets of this type contain a destination address composed entirely of 1s, which is 255.255.255.255. This is referred to as limited because routers or gateways never pass on these sorts of broadcast packets. They are intended only for a single network segment. Limited broadcasts are mostly used when computers boot so they can obtain DHCP leases or otherwise configure their network interfaces.

Smurf Attack (Using Broadcast Addresses)

- Smurf sends spoofed ICMP echo requests (also known as "ping") to a network's broadcast address
- The spoofed machine gets many, many responses, consuming most or all of its bandwidth
 - An attacker's one packet can be multiplied many times to overwhelm a network connection



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Smurf attacks utilize an amplification effect where one packets causes a large number of replies. With a Smurf attack, an attacker sends out a packet to the broadcast address of some network. All of the machines on that network will respond to the ping. However, the attacker will send the ping with a spoofed source IP address of the victim. Therefore, all responses to the ping will be sent to the victim machine, and not back to the attacker. The network that responds to the broadcast address is called the Smurf amplifier. A Smurf amplifier is an innocent set of machines on a single network segment that are configured to respond to a directed broadcast message. The attacker first finds a set of systems on the Internet that will respond to a network broadcast.

By spoofing the address of the victim, the attacker sends a series of broadcast pings to the network address. The attacker sends packet after packet to the broadcast address of the Smurf amplifier. For each incoming packet, every system on the Smurf amplifier network sends a response to the victim machine. The result is an amplifying effect.

Two Addresses

- At a minimum, a computer has two addresses
- MAC address (Layer 2):
 - 48-bit address (12 hexadecimal digits)
 - First half vendor code (00:00:0c - Cisco)
 - Used to determine the next hop
- IP address is configurable (Layer 3):
 - 32-bit address
 - Part network and part host
 - Configured by user
 - Dictated by location
 - Used to determine the path

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Two Addresses

It is important to remember that network interfaces have two addresses associated with them, namely a hardware address and a software address.

A hardware address is the Data Link Layer (Layer 2) address associated with the network interface. If the network is a Frame Relay network, for example, the hardware address is a 10-bit data link connection identifier (DLCI). If the host is attached to an IEEE LAN (such as 802.3/Ethernet), the hardware address is a 48-bit media access control (MAC) address. MAC addresses are uniquely allocated to each network interface card (NIC) and are not meant to change.

A MAC address typically is written as 12 hexadecimal digits grouped in pairs (bytes): 00-00-0c-34-17-a3 is an example of a typical MAC address. The first 24 bits of the address contain a vendor code, and the second half of the address is a unique number assigned by the vendor. MAC addresses generally are burned into NICs during the manufacturing process. Cisco Systems' vendor code, for example, is 00-00-0c, and Sun Microsystems' vendor code is 08-00-20. Readers can look up the MAC vendor codes at <http://standards.ieee.org/regauth/oui/index.shtml>.

A software address, such as an IP address, is the Network Layer protocol address. This address will be specific to the Network Layer protocol and actual network to which the host is attached. If the computer supports multiple Network Layer protocols, the NIC will have multiple software addresses.

An IP address, as discussed earlier, is 32 bits or 4 bytes in length and is usually written in dotted decimal format, such as 10.5.10.37. An IP address is hierarchical for routing purposes; the first part is the network identifier (NET_ID) and the second part is the host identifier (HOST_ID). Historically, IP used classful addresses, where the Class A, B, or C NET_ID was 8, 16, or 24 bits long, respectively.

Today, classless addressing should be assumed, where a variable-length subnet mask indicates the number of bits in the NET_ID.

On a final note, recall that every IP interface has an IP address and a subnet mask. Each IP device is also configured with the IP addresses of the name servers and default gateway (that is, the router to use if the device does have reason to send the packet elsewhere).

MAC and IP Addresses

- There isn't a direct relationship between the two addresses
- Given one address, a computer must send out a packet to find the other address
 - ARP (Address Resolution Protocol)
 - Given an IP, determine the MAC address

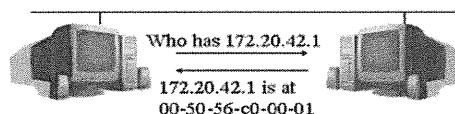
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

MAC and IP Addresses

Both direct and indirect routing require the sender to determine the MAC address of another device on the network, even though the sender knows only the IP address of the intended destination device. There is no real relationship between these two addresses; that is to say you cannot tell which IP address belongs to which MAC address by just looking at it. The Address Resolution Protocol (ARP) is used to map IP addresses to data link addresses.

ARP allows a host to find the MAC address of another host based upon the IP address, while Reverse ARP (RARP) allows a host that knows its own MAC address to query a server for its own IP address.

Address Resolution Protocol (ARP)



The sender broadcasts a packet with 42.1's IP address and asks it to respond with its physical address.

0	16	31
HARDWARE TYPE	PROTOCOL TYPE	
HLEN	PLEN	OPERATION
SOURCE MAC		SOURCE MAC
SOURCE MAC		SOURCE IP
SOURCE IP		TARGET MAC
TARGET MAC		TARGET MAC
TARGET IP		TARGET IP

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Address Resolution Protocol (ARP)

ARP, described in RFC 826, is the scheme used by one host on a LAN to determine the MAC address of another host on the same LAN. There are three scenarios in which ARP will be used: a host looking for the MAC address of another host on the LAN; a host looking for the MAC address of the default gateway; or a router looking for the MAC address of a host or another router on a LAN. The process is very straightforward.

The ARP request is a layer 2 broadcast and looks something like this: "Who has 172.20.42.1?; tell 172.20.42.2." The 172.20.42.1 machine should reply back to 172.20.42.2 with its MAC address. Now the original machine will place this entry in its ARP cache in case it needs it in the near future.

This slide shows the format of an ARP message. The field lengths shown here are consistent for Ethernet hardware addresses and IP software addresses; the names of the fields (in parentheses) are taken from RFC 826:

- Hardware Address Type (ar\$hrd): A 2-byte field specifying the type of hardware (i.e., MAC or Data Link Layer) address, such as Ethernet, IEEE 802, Frame Relay, or ATM. The value 1 (0x00-01) indicates Ethernet.
- Protocol Address Type (ar\$pro): A 2-byte field specifying the type of protocol (i.e., software or Network Layer) address, such as IPv4, IPv6, X.25, or Banyan VINES. When Ethernet is the hardware address, the protocol address field value of 2048 (0x08-00) indicates use of IPv4.

Note: Data Link frames carry data packets in the information field. In Ethernet II, the field indicating the higher layer protocol is called the EtherType. An EtherType value of 0x0800 indicates IP. ARP is a protocol independent of IP. ARP messages are carried directly in the LAN frame, using an EtherType value of 0x0806.

- Hardware Address Length (ar\$hln): A 1-byte field indicating the number of bytes in the hardware address (denoted N). For Ethernet's 48-bit address, this field's value is 6 (0x06).
- Protocol Address Length (ar\$pLn): A 1-byte field indicating the number of bytes in the protocol address (denoted M). For IP's 32-bit address, this field's value is 4 (0x04).
- Operation (ar\$op): A 2-byte code indicating the function or type of this message. Values 1 (0x00-01) and 2 (0x00-02) refer to an ARP Request and ARP Reply, respectively. Other options include RARP Request (3, 0x00-03), RARP Reply (4, 0x00-04), InARP Request (8, 0x00-08), InARP Reply (9, 0x00-09), and ARP-NAK (10, 0x00-0a; used only with ATMARP).
- Source Hardware Address (ar\$sha): An N-byte field containing the hardware address of the sender of this ARP packet.
- Source Protocol Address (ar\$spa): An M-byte field containing the protocol address of the sender of this ARP packet.
- Target Hardware Address (ar\$tha): An N-byte field containing the hardware address of the target of this ARP packet.
- Target Protocol Address (ar\$tpa): An M-byte field containing the protocol address of the target of this ARP packet.

Domain Name System (DNS)

The student will have a high-level understanding of the Domain Name System architecture.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Domain Name System (DNS)

This section intentionally left blank.

Domain Name System (DNS)

- Static host tables
- Converting IP addresses into hostnames
- DNS hierarchy
- Types of DNS queries

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Domain Name System (DNS)

Now you know a lot about how programs and computers tell each other with whom they would like to communicate. How, then, do people tell computers? Imagine a world where your e-mail address was just a bunch of random numbers. It is not easy to remember eric@216.139.129.11. People just don't remember numbers very well. We tend to remember words more easily. Modern computing systems have a few different ways to convert the human-friendly yahoo.com name into the IP address 216.139.129.11. Let's look now at two of the most popular, static host tables and the Domain Name System (DNS).

Two Ways to Resolve IPs

- Static host tables: Each host has all the IP/hostname pairs it wants to resolve in a file.

```
$ cat /etc/hosts
```

```
127.0.0.1 loopback
```

```
172.20.1.41 relay relay.sans.org
```

```
172.20.31.19 goo goo.sans.org
```

```
...
```

- Domain Name System: A distributed, hierarchical database maps IP addresses to hostnames.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Two Ways to Resolve IPs

The Internet used to be a much smaller place than it is now. Twenty-five years ago, the number of hosts on the Internet was much smaller, not millions.. Still, with all those hosts, users could not be expected to remember and exchange IP addresses whenever they wanted to send an e-mail message or transfer a file. The designers of the Internet (if there could ever be a single group that could be said to have designed the Internet) came up with a system that is still in wide use today. Their idea was that humans should be able to refer to individual computers by name rather than IP address. They implemented this idea by keeping a text file on each machine that contained the IP address and name of every other computer with which the host would ever want to talk.

On Unix machines (the majority of the Internet at that time), this file was named /etc/hosts. In fact, InterNIC, the organization that for decades served as the central clearinghouse for Internet naming information, used to keep a single host file that listed every computer connected to the Internet. System administrators would configure their systems to download a fresh copy every so often in order to stay up-to-date. Although this worked for a time, the system eventually became unwieldy, as the number of Internet hosts grew. InterNIC became a single point of failure for the whole system.

Static host tables are still in common use today. Practically every machine with a TCP/IP stack implements a host file. Unix and Linux systems still keep them in /etc/hosts. Windows XP/Vista/Windows 7 and 2003/2008 machines typically store this file in %systemroot%\system32\drivers\etc\hosts, with a second similar file in the same directory called lmhosts that contains additional mappings for NetBIOS-to-IP address translations. These files rarely list every host on the local network anymore, let alone the hosts on the entire Internet. Today, most hosts use the Domain Name System (DNS) and can use the host file if they do not want to use DNS.

Domain Hierarchy

- Protocol for resolving IP addresses to domain names (and back again)
- Hierarchical system of domain names
- Root-level servers for top-level domains: .com, .net, .org, .info, .biz, .aero, .coop, .museum, .name, .pro, .gov, .edu, and .int
- Country codes: .us, .ca, .au, .gb, .jp, .cn, .hk, and so on

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

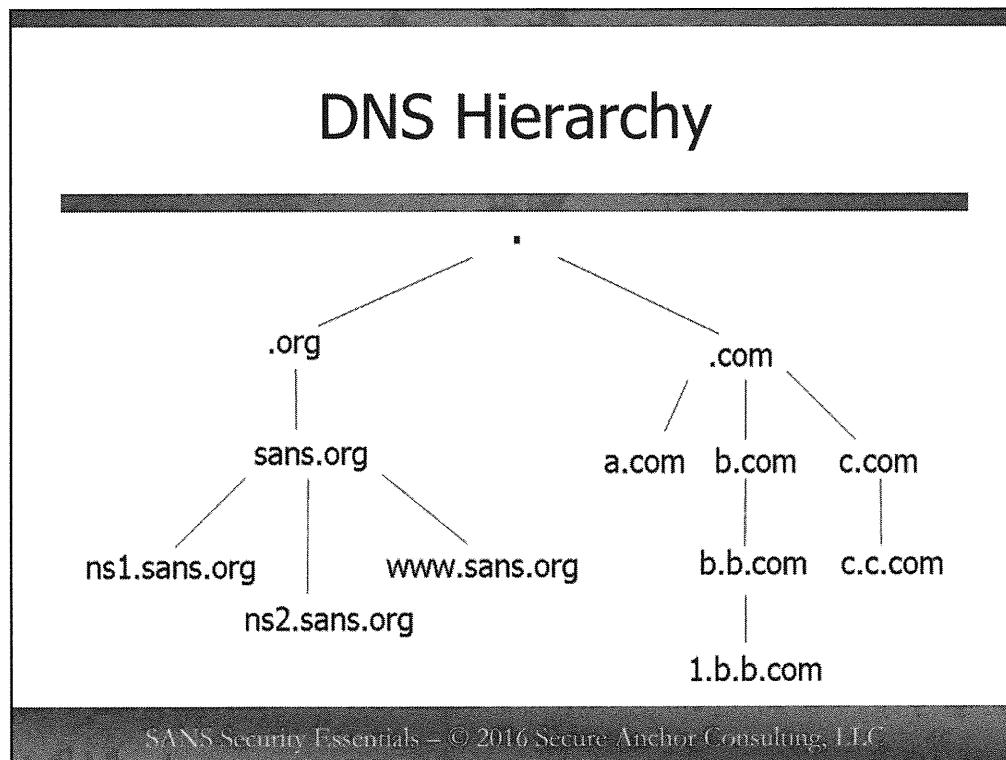
Domain Hierarchy

There are two types of top-level domains: generic and country code plus a special top-level domain (.arpa) for Internet infrastructure. Generic domains were created for use by the Internet public, while country code domains were created to be used by individual countries as they deemed necessary. Following are links to more information about each type of top-level domain, including contact information or links to their registration services.

- Country code domains (.uk, .de, jp, .us, and so on)
- Generic domains (.aero, .biz, .com, .coop, .edu, .gov, .info, .int, .mil, .museum, .name, .net, .org, and .pro)

<http://www.iana.org/domain-names.htm>

Like a host table, DNS is a method for mapping human-readable hostnames into IP addresses and back again. Unlike the host file, however, there's no single place that stores every name and IP address on the Internet. DNS was one of the first worldwide, distributed databases. Each organization connected to the Internet maintains (or pays their ISP to maintain) a small part of this database containing information about its own hosts. DNS clients, or resolvers, on other hosts can query this database and resolve hostnames properly, whether the servers are in the next room or the next continent. DNS is truly one of the most important technologies that make the Internet work.



DNS Hierarchy

Because of their importance in making the Internet work, you are, no doubt, familiar with DNS names, even if you do not know it yet. DNS is a top-down, hierarchical naming system comprised of domains. There are special cases of domains, such as top-level domains and sub-domains, but the general term domain applies to all types. The domains are structured in a hierarchy like a tree; the top-level of the tree is called the root or the top-level domain. There are a handful of such top-level domains: .com, .biz, .gov, and .edu. It is important to note that .gov, .mil, and .edu are typically reserved for use in the United States. In addition, most countries have their own, such as .us for the United States and .uk for the United Kingdom.

Within each top-level domain, there are a number of other domains. If the top-level domain is .com, some valid sub-domains might include ibm.com, yahoo.com, or microsoft.com. This slide represents this hierarchy visually. Each level of the tree can be further broken up into either individual hosts (samurai.sample.org) or more sub-domains (netlab.sample.org). There can be any number of levels, but you will rarely see more than four or five.

Each registered domain or sub-domain has one or more DNS servers that are authoritative for that domain. An authoritative DNS server contains the IP address to hostname mappings for all servers in that domain, as well as the list of DNS servers that serve any sub-domains beneath the domain. To add or delete servers from a domain, the domain administrator will update the DNS records stored in the authoritative servers.

Types of DNS Queries

- **Gethostbyname:** forward lookup
 - Maps fully qualified domain name (FQDN) to IP address
 - For example, maps www.sans.org to its IP address
- **Gethostbyaddr:** reverse lookup
 - Maps IP address to FQDN

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Types of DNS Queries

When a DNS client wants to look up an address, it contacts a local DNS server that the administrator has configured. If the request is for a host that the server has in its database, it supplies the requested information to the client, and the request is considered authoritative. If the server does not have the requested information, it generally refers the client to another server that might. The first such referral usually is to one of several root name servers.

The root name servers never know the final answers to the queries. Instead, their job is to refer DNS clients to the proper server for the top-level domain about which they are being queried. For example, if the client asked for www.sans.org, the root name server would refer it to a top-level name server handling the .org domain; and the client would begin another query with that server.

The top-level name servers usually do not know the answer to any queries either. They do almost the same thing as the root name servers. Instead of returning an answer, they refer DNS clients to other servers that might know the answer. In our example, the .org name server will refer the client to the name server for the sans.org domain.

When the client queries sans.org's name server, it is quite likely that it will receive an authoritative reply instead of a referral. If so, then the client has the information it needs and can move on to making the actual connection. In order to avoid having to query DNS every time it needs to contact www.sans.org again, the resolver usually caches the reply.

The DNS servers set a Time-To-Live (TTL) value on the records in their databases that tell DNS clients how long it is safe to cache the information. Well-known sites that rarely change usually have long TTLs to avoid overloading their DNS servers with repeated requests. Sites that change frequently usually have short TTLs, so clients won't keep trying to connect to invalid IP addresses.

Recursive Queries

Clients can request another type of query and ask the server to do all the hard work for them. Clients can make recursive queries to their local name servers, which request that the name server itself handle all the referrals and not bother the client with anything but the final answer. Although this method places more load on the DNS server, it is advantageous because it allows the server to cache DNS queries for many hosts. After one host has looked up www.sans.org, the server can then answer the same query from its cache more quickly for other clients. Answers that come from cache are referred to as non-authoritative answers because a DNS server that does not house the actual database for that domain supplied them.

Making a DNS Query

You do not have to rely on your applications to make DNS queries on your behalf. If you want to query DNS yourself, you can use a DNS client utility like nslookup. Try the following command under Unix or XP/Vista to look up information about one of Yahoo's Web servers:

```
$ nslookup www.yahoo.com
```

Server: cache04.ns.uu.net

Address: 198.6.1.5

Name: www-real.wa1.b.yahoo.com

Address: 209.191.93.52

Aliases: www.yahoo.com, www.wa1.b.yahoo.com

Within a fraction of a second, the system tells you the information you requested. This is known as forward resolution or a forward lookup. But that is not all DNS is good for. What if you have the IP address and need to know the name that goes with it? You need a reverse lookup. Try this:

```
$ nslookup 216.109.118.72
```

Server: cache04.ns.uu.net

Address: 198.6.1.5

Name: ha8.ge-2-19.bas-1-con.ac2.yahoo.com

Address: 216.109.118.72

As you can see, it is quite easy to query DNS yourself. Internally, the DNS resolver API uses the gethostbyname function to perform forward lookups and the gethostbyaddr function to do reverse lookups. If you are programming a network application, you need to be familiar with these calls; but for most of us, the nslookup command is good enough for anything we need to find.

DNS Security

- **Defenses:**
 - Keep DNS software up-to-date
 - Distribute authoritative DNS servers
 - Limit zone transfers
 - Limit recursive lookups
 - Register with reputable registrars
 - Split DNS

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

DNS Security

DNS was conceived at a time when the Internet was relatively young. At that time, experts believed it would be used principally for academic information exchange. In those days nobody had imagined the size or the scope of the modern Internet, let alone that it would become a media for e-commerce and mission-critical communications. DNS is vital to the operation of the Internet, and unfortunately, DNS was not designed with a great deal of security in mind. As a result, DNS is susceptible to a number of different attacks. The rest of this section describes some of the attacks that have been common in the recent past.

Cache Poisoning

Lately, cache poisoning attacks have been in vogue. DNS cache poisoning attacks involve returning extra data along with the results of a query. This extra data contains invalid information, which on vulnerable DNS servers will be written to the DNS cache, thus poisoning the DNS cache for the server. The end result is that any traffic for a server with a poisoned entry could be redirected to a server the attacker controls.

The best defense against cache poisoning is keeping your DNS software updated to the most recent version and keeping patches up-to-date, decreasing the likelihood that the DNS server is susceptible to vulnerabilities that permit the cache poisoning to succeed.

Denial of Service

Denial of service attacks on DNS servers involve flooding legitimate DNS servers with a large number of queries. If a sufficient volume of queries is received, they will overwhelm the DNS server so that it cannot respond to legitimate queries. This effectively makes servers in the domain served by the DNS server unavailable.

There is no good way to completely mitigate against a persistent denial of service attack; however, there are commercial denial of service mitigation appliances that will shed traffic in a denial of service scenario. The more effective approach is to geographically disperse your authoritative DNS servers and/or host them at providers with big network connections so the success of the attack will be limited.

Footprinting

Footprinting involves using DNS data to learn about the servers in a network. This can be done by requesting zone transfers against improperly configured DNS servers or by performing reverse DNS lookups against an entire network range. The gathered information can be used to formulate attacks against servers in the address space.

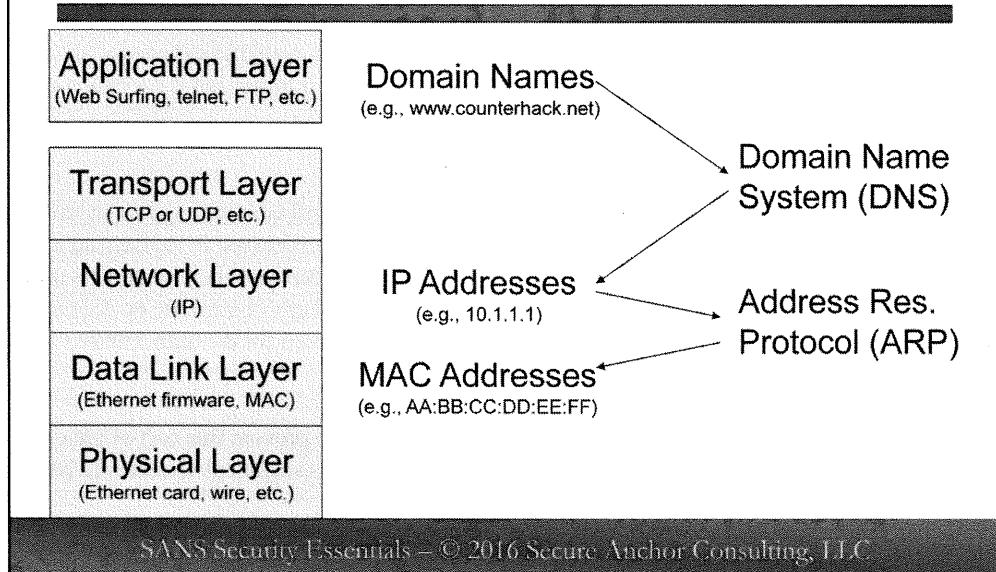
There are two principal defenses against footprinting. The first is to limit zone transfers to only DNS servers who legitimately require them. The second is to limit the DNS information available externally to only the information for your Internet accessible servers. This limits the amount of information the attacker can gather and can be done through split DNS.

Registration Spoofing

Registration spoofing is not an attack directly against the DNS servers or infrastructure, but a social engineering attack against the registrar of the domain. One of the pieces of information involved in registering a domain is the DNS servers that are authoritative for the domain. If the attacker can convince the registrar to alter the registration record for the domain to point to a DNS server under the attacker's control, he or she can redirect all DNS requests for that domain to his or her DNS server and all traffic destined for the authoritative servers in the compromised domain to servers under the attacker's control.

Most of the larger, more mainstream registrars have procedures in place to prevent this sort of attack. Register your domains with the larger registrars, or at least ensure your registrar has sufficient checks and balances in place to prevent registration spoofing.

How the Layers Fit Together



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC.

Just to review. First, remember how DNS maps domain names (Application Layer) to IP addresses (Network Layer). We also use ARP to map IP addresses (Network Layer) to MAC addresses (the Data Link Layer hardware address). This diagram shows how all of those pieces fit together.

IPv6

The student will have a high-level understanding of the IPv6 protocol.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IPv6

This section intentionally left blank.

IPv6

- IPv4 accommodates 4.2 billion unique 32-bit addresses
- New technology growth requires more address space
- IPv6 is designed to meet addressing growth:
 - 128 bits accommodate 340 undecillion addresses (7 addresses for each atom of every human)
 - Offers greater flexibility in allocating addresses

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IPv6

The IPv6 protocol was designed to supersede IPv4 addressing while supporting the growth of the Internet. Although the IPv4 protocol accommodates 4.2 billion unique IP addresses with a 32-bit address, the allocation of IP addresses on the Internet was not completed in the most efficient manner, leaving a shortage of available IP addresses. With deployment of technologies such as NAT and CIDR, the Internet continued its growth, but was still somewhat limited without the widespread availability of globally unique IP addresses. New technologies such as mobile phones and PDAs connecting to the Internet have increased demand for addresses, as has the spread of Internet technology to populous countries such as China and India. As a result, a new mechanism was needed to accommodate continued growth and adoption of Internet-connected technology.

The IPv6 protocol was designed to meet these growth demands, expanding the address size from 32 bits to 128 bits. A 128-bit address is approximately 340 undecillion addresses or 340,282,366,920,938,463,463,374,607,431,768,211,456. With this many unique addresses, the IPv6 protocol can accommodate 7 unique IP addresses for each atom in every human on earth.

Of course, all of our atoms don't need that many IP addresses. Instead, the sheer volume of available IP addresses provides for more flexible deployment of address space on the Internet. For example, ISPs will be able to geographically assign IPv6 prefixes to different parts of the world, allowing for the simplified routing of traffic on the Internet. Organizations can obtain an IPv6 prefix with sufficient available addressing to accommodate all present and future addressing needs.

IPv4 Versus IPv6

IPv4	IPv6
32 bits addresses 4.2 billion addresses	128 bits 340 undecillion addresses
No authentication	Provides authentication of endpoints
Encryption provided by applications	Support for encryption in protocol
Best effort transport	Quality of Service (QOS) features provided in the protocol

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IPv4 vs. IPv6

When IPv4 was conceived, it was designed to support academia and scholarly research. Nobody had conceived that the Internet would be supporting e-commerce and mission-critical and real-time sensitive applications. With IPv6, consideration was made for authentication, security of the communication, and quality of service features.

IPv6 incorporates aspects of IPsec to provide authentication of endpoints and encryption of the packets in transport. Also included in IPv6 are Quality of Service (QOS) features that will permit real-time sensitive applications such as VoIP and interactive media to take priority over less critical packet streams.

IPv6 Features

- Extended address space:
 - Route aggregation, improved delegation/management, hierarchy
- Autoconfiguration support
- Support for IPv6 over IPv4 (tunneling)
- Support for IPv4 over IPv6 (translation)
- Flexible embedded protocol support
- Support for authentication of endpoints
- Support for encryption

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IPv6 Features

A key feature of IPv6 is the expansion of address space, permitting route aggregation on core Internet routers through geographic address space allocation, improving delegation and management of addresses to organizations and ISPs alike, as well as providing hierarchical distribution of address space that makes troubleshooting and Internet routing simpler.

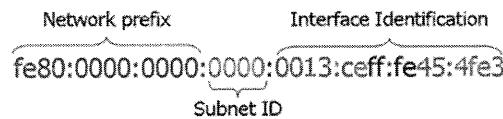
Another valuable feature of IPv6 is support for addressing autoconfiguration. Anyone who has been responsible for manually assigning IP addresses to hosts understands that this is a problematic and cumbersome process. With 128 bits of address space, it becomes possible to use the globally unique MAC addresses on all network cards as IP addresses. In this way, administrators can simply introduce a new node to an IPv6 network without manually specifying an IP address; the IP address is configured automatically based on the local MAC address and advertisement information from the default gateway on the network.

During the transition process between IPv4 and IPv6, it is possible to establish IPv6 tunnels over the existing IPv4 Internet using IPv4 Protocol 41 or one of several tunneling protocols such as AYIYA (Anything In Anything) or Teredo (Tunneling IPv6 over UDP through NAT). Further, it is also possible to continue supporting IPv4 traffic on an IPv6 backbone using gateway services that translate IPv4 packets into an IPv6 format.

Another significant change in the IPv6 protocol is the use of a fixed IP header. While the IPv4 header can expand to include additional information such as strict or loose source routing, the IPv6 protocol has a fixed header length of 40 bytes. In order to accommodate additional flexibility in the protocol, IPv6 introduces a next header field that indicates what the embedded protocol contained in the packet payload is. This is similar to IPv4's embedded protocol field, but unlike this field, the next protocol can include multiple embedded protocol fields, one right after another. Currently supported IPv6 next header protocols include the encapsulating security protocol (ESP) and authentication header protocol (AH) for IPSec, the destination options header to specify processing options at the destination system, and upper-layer protocols such as UDP, TCP, and ICMP.

IPv6 Addressing

- Address is 128 bits
- Addresses specified in hex, colon-delimited, 32 hex digits
- Divided in three portions:
 - Network prefix (48 bits): Defines the organization
 - Subnet ID (16 bits): Used internal to the organization for subnetting
 - Interface ID (64 bits): Defined from the MAC address of the NIC (autoconfiguration)



- Groups of repeating 0000s can be simplified with ":"
fe80::13:ceff:fe45:4fe3

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IPv6 Addressing

Because of the longer address space, changes have been made to how IP addresses are represented from IPv4. Where it is simple to specify a 32-bit address in dotted-decimal notation, remembering an address that is four times longer in the same format can become overwhelming quickly. In IPv6, IP addresses are represented using hexadecimal notation, with values separated by colons instead of dots. For the foreseeable future, many IPv6 addresses will include strings of four repeating 0s ("0000"), which can be condensed to just a single colon to represent one or more groups of 4 zeroes.

IPv6 addresses are broken up into three major sections: the network prefix, the subnet ID, and the interface identification section.

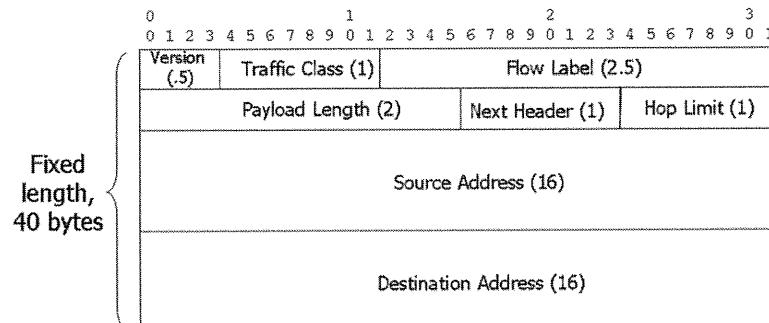
Network prefix: The network prefix is represented in the first 48 bits (6 bytes) of the IPv6 address. This is the address portion that is allocated to organizations that need to address IPv6 clients or to preserve other network functionality. Some fixed network prefix allocations include "fe80::" for local network use; "ff00::" for multicast traffic; "2001::" for large ISP inter-domain routing; and "2002::" for IPv6-to-IPv4 gateway networks.

Subnet ID: The subnet ID is configured according to the addressing needs of the organization. For flat IPv6 networks, this value will usually be "0000", but can be any value selected by the organization that has been given the network prefix.

Interface identification: The interface identification section uniquely identifies the IPv6 node. With IPv6 autoconfiguration, the MAC address of the client populates the interface identification portion of the IPv6 address. Because a MAC address is a 48-bit value and the interface identification portion of the IPv6 address is 64 bits, the MAC address is expanded to fill the space by converting it to the Extended Unique Identifier (EUI) format specified by the IEEE. The EUI expansion takes the first three octets of the MAC address, appends the constant value "ff:fe", and then appends the last three bytes of the MAC address to form the interface identification portion of the IPv6 address.

As an example, if a small, flat network uses private addressing, it would most likely use fe80:0000:0000 as the network prefix. If the network adapter of the host has a MAC address of 00:13:ce:45:4f:e3, converting the MAC to EUI-64 notation, the interface identifier would become 0013:ceff:fe45:4fe3. Putting it all together, IPv6 autoconfiguration would assign an IPv6 address of fe80:0000:0000:0013:ceff:fe45:4fe3. By convention, in IPv6, groups of consecutive zeros can be replaced with "::", thus, permitting us to write the address as fe80::13:ceff:fe45:4fe3

IPv6 Header



Traffic Class+Flow Label provide QoS. Next Header indicates embedded protocol data. Hop Limit prevents routing loops.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IPv6 Header

In order to accommodate the changes in the IPv6 protocol, the header information has changed by removing superseded functionality from the IPv4 header and introducing some new fields:

- Version: 4 bits. The version field indicates the packet is IPv6 and is always a 6.
- Traffic class: 1 byte/8 bits. The traffic class field is used to specify the priority of the packet for QoS.
- Flow label: 20 bits. The flow label field is used for QoS management to convey special handling functions for the packet.
- Payload length: 2 bytes/16 bits. The payload length field specifies the length of the packet in a quantity of bytes.
- Next header: 1 byte/8 bits. The next header field specifies the next encapsulated protocol in the payload of the packet. The values that are assigned to IPv4 embedded protocols, such as TCP, UDP, and ICMP, are forward-compatible with the IPv6 next header field.
- Hop limit: 1 byte/8 bits. The hop limit field is used to prevent routing loops by decrementing the hop limit value at each router. This is similar to the TTL field used in the IPv4 header.
- Source address: 16 bytes/128 bits. This field is the source address of the IPv6 station transmitting the packet.
- Destination address: 16 bytes/128 bits. This field represents the destination or recipient of the IPv6 packet.

Summary

- Protocols
- OSI and TCP/IP models
- IPv4
- Addressing
- Domain Name System (DNS)
- IPv6

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

First, you learned a bit of theory about protocols and their organization into stacks. The OSI reference model is a widely used standard. Despite that, the number of people who really understand it is probably lower than it should be. If you remember the OSI protocol stack, you'll do well. The same goes for TCP/IP. Most people tend to view TCP/IP as a single, monolithic protocol; but those of us in the know realize that this isn't so. In fact, you could say that its strength comes from its layered approach to making different protocols work together smoothly.

We took a quick look at DNS and how it makes the Internet easier for people to use. We looked at how DNS is a distributed hierarchical database and the different types of DNS queries. We also looked at some of the potential security pitfalls and how to avoid them. Finally, we finished by looking at IPv6.

Module 4: TCP, UDP, and ICMP

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 4: TCP, UDP, and ICMP

This section intentionally left blank.

TCP, UDP, and ICMP

SANS Security Essentials I: Networking Concepts

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

TCP, UDP and ICMP

In the last module, you learned about the basic building block of internetworking: the Internet Protocol (IP). We showed you how it forms one layer of a standardized network stack that, along with the other layers, allows two computers to exchange data over a network, even if they run different operating systems or are built by different vendors.

Objectives

- Layer 4
 - Transmission Control Protocol (TCP)
 - User Datagram Protocol (UDP)

- Layer 3
 - Internet Control Message Protocol (ICMP)

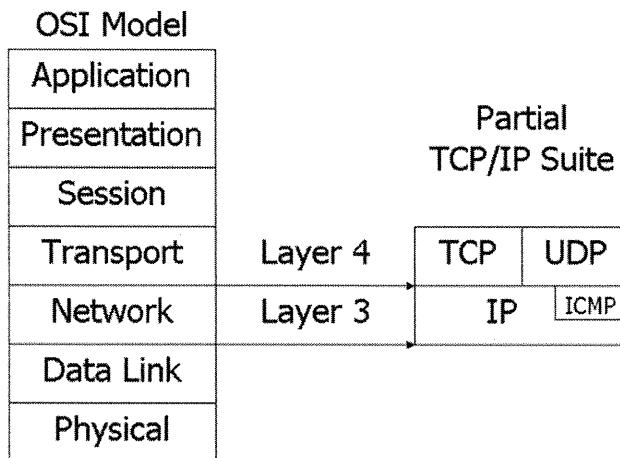
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

IP never was intended to stand by itself. Although it is possible to write programs that talk directly to the Network Layer (IP), it is rarely done, except in certain limited cases, such as network testing tools or hacking utilities. Most programs do not want to have to deal with the level of complexity that speaking directly to the Network Layer brings and instead are written to make use of higher-level protocols residing in the Transport Layer. For IP, these are the User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP). We introduced these protocols in Module 2 because you can hardly discuss IP and IP stacks without mentioning them. In this module, however, you learn more about them and how they work. We also examine the Network Layer's Internet Control Message Protocol (ICMP). IP relies on ICMP for network status messages and error reporting. ICMP messages are quite common on any IP network, so ICMP is just as important as TCP and UDP from a security standpoint.

We end this module by demonstrating how IP, ICMP, and UDP are used in the program called traceroute to identify the path traffic takes to get from one endpoint to another.

IP Protocols and the OSI



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IP Protocols and the OSI

Before we begin our discussions of TCP, UDP, and ICMP, we should review their relationship to the Open Systems Interconnection Reference Model (OSI Model).

The Network Layer, commonly referred to as Layer 3 of the OSI Model, is responsible for determining routes to be taken between two network devices and for handling flow control, segmentation/desegmentation, and error control functions. ICMP performs a subset of these functions and works with the IP protocol, along with some other sub-protocols of the TCP/IP Suite, to complete this functional portion of the OSI Model.

The Transport Layer, commonly referred to as Layer 4 of the OSI Model, is responsible with the transmission of data between the two end-point systems involved in the communication. Issues related to reliability and cost-effective data transfer belong to this layer. In the TCP/IP suite, TCP and UDP are the most used and well-known protocols that function at this layer.

Transmission Control Protocol (TCP)

The student will understand the structure and purpose of TCP and the fields in a TCP datagram header.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Transmission Control Protocol (TCP)

This section intentionally left blank.

TCP

- Most commonly used transport protocol today
- Connection-oriented communications
- Provides guaranteed packet delivery
 - Additional overhead required to track packet delivery

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

TCP (Transmission Control Protocol)

TCP is the most commonly used Transport Layer protocol today. It establishes a virtual connection, often referred to as a session, between the hosts. The protocol is designed to provide reliable connections over possibly unreliable networks. Unlike UDP, which blindly sends datagrams and hopes they arrive, TCP can guarantee that the packet will arrive or at least that it will notify you of a problem. Because of this guarantee, TCP often is a network programmer's protocol of choice. It is probably the easier of the two protocols to program for, because most of the error handling is down inside the Transport Layer and out of sight from the application code. TCP is especially useful for any application in which there are more than one or two network hops between two computers, because more hops equals more chances for errors to be introduced into the communication.

Most of the Internet protocols you use everyday are based on TCP. Some examples include HTTP (HyperText Transfer Protocol), used by Web servers and browsers); FTP (File Transfer Protocol), used to transfer files to and from servers); or POP3 (Post Office Protocol version 3), which is used to download e-mail).

TCP Uses

- Offers flow control to handle network congestion
- Allows for transmission of larger amounts of data per packet
- Guaranteed delivery of transmitted data is more important than speed
- Offers better protection against spoofing attacks
- Common TCP ports: FTP Data (20), FTP (21), Telnet (23), SMTP (25), DNS (53), Finger (79), HTTP (80), POP (110), and HTTPS (443)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

TCP Uses

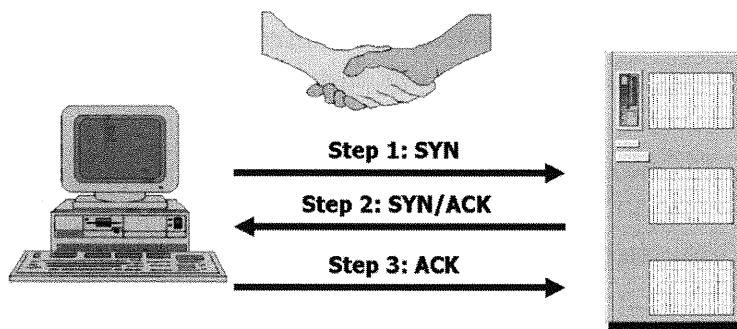
There are many reasons that TCP is more popular than UDP. In fact, we mainly have TCP to thank for the fact that the Internet is as reliable as it is. Without the congestion control capabilities of TCP, it is probable that systems attached to the Internet would send so much data that the accumulation of all hosts on the Internet would cause a fundamental failure due to inadequate bandwidth capabilities on the Internet backbone. Because of this congestion control, systems are able to self-regulate their bandwidth usage in order to adapt to changing conditions of bandwidth capacity.

Another important feature is that TCP allows for more data to be sent in a single packet than UDP. This is especially helpful at reducing the overall processor overhead when transmitting large amounts of data between two hosts.

Likely, the biggest reason that TCP is the most popular transport protocol is the built-in capabilities that work to ensure all of the data for a particular session is received. This is achieved by error detection and efficient methods to resend missing data and reconstruct the data in the order in which it is intended with confidence that the entire data stream has been received.

Application developers generally prefer TCP to UDP due to a reduced requirement that higher-level applications validate that information was received in the same order and without error in transmission when compared with the way it was sent. Because these details are handled as part of the TCP mechanisms, developers can put this concern aside and spend more time working on the core functionality of the application.

Establishing a TCP Connection



A TCP connection is established by a three-way handshake in which ISNs (initial sequence numbers) are exchanged.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Establishing a TCP Connection

TCP connections are established using the three-way handshake. This procedure is required before the two hosts can exchange any data. In the three-way handshake, segments are often named after the flags they have set. Therefore, a segment containing a lone SYN segment is also called a SYN, and a lone ACK segment is called an ACK. A segment with both SYN and ACK is called a SYN-ACK.

The client initiates the three-way handshake by sending a SYN to signal a request for a TCP connection to the server. Then, if the server is up and offering the desired service, it can accept the incoming connection and respond to the SYN. The response consists of both an acknowledgment of the client's initial connection request (the ACK flag is set) and a connection request of its own (the SYN flag is set), together in a single packet (a SYN-ACK).

Finally, after the client receives the SYN-ACK, it sends a final ACK to the server. After the server receives the ACK, the three-way handshake is complete, and the connection has been established. The two servers can then exchange data.

After a connection is established, the ACK flag is set for every packet. As a result, the presence of the ACK can indicate whether a connection has been established or not. In fact, simple packet filters allow all packets with ACK set and assume that they are part of an established connection. It is trivial to circumvent such a filter by crafting a packet with the ACK bit set. This technique is often used to probe a network behind a filtering device and called an ACK scan.

To minimize traffic, ACKs are "piggy-backed" (as frequently as possible) onto packets containing data, as opposed to sending a packet with just an ACK. The ACKs confirm to the client and server that both ends are still using the connection.

TCP Header

0	16	31
TCP SRC Port (2 bytes)	TCP DST Ports (2 bytes)	
Sequence Number (4 bytes)		
Acknowledgement Number (4 bytes)		
Offset HL (4)	Reserved (6)	Flags (6)
		Window (2 bytes)
Checksum (2 bytes)	Urgent Pointer (2 bytes)	
Options (Optional – variable length)		
Data (variable length)		



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

TCP Header

Because TCP is a much more heavyweight protocol than UDP, it requires a much larger header. The normal TCP header is a 20 bytes! Because most TCP implementations also specify options, it can grow even larger. From a security standpoint, some of these fields are more important than others. Let's take a look at some of the key elements of the TCP header.

TCP Source Port - 2 Bytes and TCP Destination Port - 2 Bytes

The source port indicates the port on the sender that the receiver should reply to, while the destination port indicates the service on the receiver to which the packet should be delivered. Valid port numbers are 1 through 65,535.

Sequence Number - 4 Bytes

The sequence number is a value used to indicate the first data octet for the segment being sent.

Acknowledgement Number - 4 Bytes

The acknowledgement number indicates the value of the next sequence number the sender is expecting to receive from the other party to the communication.

Offset (Header Length) - 4 Bits

This value (offset or header length) indicates the number of 32-bit words (4 bytes) that are contained in the TCP header for this packet. The minimum decimal value for this field is 5 to indicate 20 bytes of information (4 bytes * 5 = 20 bytes).

Reserved - 6 Bits

When the TCP protocol was initially created, these 6 bits were reserved for future use. Since that time, the last 2 bits in this segment have been identified for Explicit Congestion Notification (ECN) usage.

Flags - 6 Bits

These six bits are used to signal whether specific TCP flags are on or off. The flags in order are Urgent (URG), Acknowledgement (ACK), Push (PSH), Reset, (RST), Synchronize (SYN), and Finish (FIN). There are two additional bits located at the **most significant position of the flag's byte that are used for ECN or explicit congestion notification.**

Window - 2 Bytes

The window value indicates the number of data octets that the sender of this segment is willing to accept.

Checksum - 2 Bytes

This is a one's complement of the one's complement sum of all 16-bit words in the header and text for purposes of identifying data that may have been corrupted in transmission.

Urgent Pointer - 2 Bytes

The urgent pointer value is used only in conjunction with the URG flag being set. This field points to an offset from the sequence number where urgent data resides.

Options (Optional) - Variable Length

TCP options are not required. If they are used, this field is occupied by a minimum of 4 bytes and will consume multiples of 4 bytes. Unneeded space will be padded with zeroes. Common TCP options include Maximum Segment Size (MSS), Windows scale, Selective ACK ok (SACK ok), Timestamp, and No operation (NOOP).

With the exception of the Options field, all of the fields in the TCP header are fixed length, meaning they will always occupy the same location in a TCP packet. The variable length of this field and the following payload or data field often require the need to perform calculations to determine where certain parts of a packet begin and end.

TCP Header - Key Fields

The TCP source and destination ports are identical to their UDP counterparts. The source port indicates the port on which the sender is listening, and the destination port indicates the port to which the packet should be delivered on the receiving side.

In the past, well-known server ports generally fell below port 1024. These ports should remain constant on the host on which they are offered. In other words, if one day you find Telnet at port 23 on a particular host, you should find it there the next day. You will find many of the older, well-established services on ports below 1024, such as SMTP (port 25). It would be impossible to assign a distinct number in this range to every well-known service now that there are so many.

Client ports, often known as ephemeral ports, are normally session source ports that are selected only for a particular connection and are then made available to be reused after the connection is freed. Ephemeral ports are usually numbered 1024 or higher; the largest possible ephemeral port is 65,535. When a client initiates a connection to a server, it selects an unused ephemeral port. For most services, the client and server continue to exchange data between the ephemeral port and the server port for the session's entirety. This pair of ports is known as a socket, and it is unique to the particular exchange. That is, there is only one connection on the Internet at any given time that has this combination of source IP and source port connected to this destination IP and destination port.

Sure, another user can connect from another source IP to this same destination IP and destination port, but that user has a different source IP and most likely a different source port. There might even be someone from the same source IP connected to the same destination IP and port; however, this user is given a different ephemeral port, thereby distinguishing it from the other connection to the same server and destination port. As an example, two users on the same host might be connecting to the same Web server. Although these two users have the same source IP, the same destination IP, and port (80), the Web server can maintain which data goes to which user because the ephemeral source ports differ.

TCP uses sequence numbers to track packets and provide reliable delivery of information. The host that is sending the data uses sequence numbers; the receiving host uses acknowledgment numbers to acknowledge the receipt of data.

TCP numbers every byte of data it sends with a unique sequence number. This allows either side of the connection to refer to specific bytes by number (that is "the 103rd byte you sent me"). A connection's Initial Sequence Number (ISN) is the first sequence number used in that connection. TCP initializes the ISN to a random or semi-random value (for security reasons, the more random the value, the better). Sequence numbers for the rest of the bytes in the connection are then derived from the ISN by incrementing it by 1 for each byte sent. The sequence number of the first byte sent always equals the ISN + 1. Therefore, if the ISN was 3003873, the 103rd byte would be sequence number (3003873 + 103) or 3003976.

Older TCP implementations used to start ISNs at 1 and increment them by a fixed number (usually 64,000) for each new connection made. More modern stacks start with a random value and increment by different random values for each connection, to keep anyone from guessing what the next valid ISN might be. The best stacks do not increment at all, and return a different pseudo-random ISN for each connection. A given connection could therefore have a lower ISN than the one before it, making it virtually impossible to guess.

The sequence does not start over again with each new packet. It continues until the connection is closed. As we just saw, if a certain packet has a sequence number of 3003873 and contains 103 bytes, the sequence number of the last byte in the packet is 3003976. The sequence number of the first byte in the next packet will be 3003977. If the connection should ever transmit enough data that this 32-bit field would be too small to contain the actual next sequence number, the count rolls over to 0 and continues.

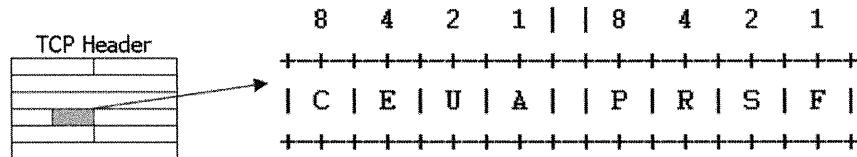
Acknowledgement numbers are closely tied to sequence numbers. TCP is required to acknowledge every byte of data that it receives. To acknowledge receipt of all data up to a certain byte, the receiver puts that byte's sequence number into this field, increments it by 1, sets the ACK flag and sends a packet back to the sender. That is, the acknowledgement number does not specify the last byte received. Rather, it specifies the sequence number of the next byte that the receiver expects. Therefore, to acknowledge byte 100, the acknowledgement number would be (ISN + 101), which is the number of the next byte in the sequence.

By sending an acknowledgement, the receiver acknowledges receipt of every byte leading up to that acknowledged byte. For example, it is not possible to indicate that you received bytes 90 through 100, but that you did not receive 85 through 90. If the receiver acknowledges byte 100, it is implicitly acknowledging all preceding bytes. If some packets arrive out of order, the higher sequence numbers are put "on hold" until all the other lower sequence number bytes arrive, and are then reassembled into a coherent stream. If the missing bytes never arrive, the sender times out waiting for them to be acknowledged and eventually sends them again, starting just after the last byte for which it received an acknowledgement.

The SYN, or synchronization bit, is used when establishing a connection and is only used in the first two exchanges of the TCP three-way handshake. The ACK, or acknowledgement bit, is used when a system is acknowledging the receipt of information. In the three-way handshake, the second and third exchanges are acknowledged.

TCP Code Bits / Flags

- Controls data flow and signal information to the receiving host



- CWR - Congestion Window Reduced
- ECE - ECN Echo
- URG - Urgent
- ACK - Acknowledgement
- PSH - Push
- RST - Reset
- SYN - Synchronize
- FIN - Finish

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

TCP Code Bits / Flags

TCP stacks need to communicate about the data they're exchanging. The catch is they cannot insert their own information into the payload because that would corrupt the data stream and might confuse the applications. Instead, the TCP protocol provides six one-bit flags that can be specified in the packet headers. Some of these are more common than others, but the unusual use of TCP flags is a good indicator of suspicious traffic, so you should become familiar with all of them.

Following are the different flags:

- CWR (Congestion Window Reduced): The CWR flag is associated with a protocol known as Explicit Congestion Notification (ECN). This bit and the one that is now assigned to ECE used to be reserved. ECN was proposed several years ago and, although the TCP and IP fields exist to support it, it has yet to catch on.
- ECE (ECN Echo): The ECE flag is also associated with ECN.
- URG (Urgent): The urgent flag is used by some applications, such as Telnet and rlogin. An application can set this bit to let the other end of the connection know that some important data is coming; but it is up to the client to decide what is urgent and up to the server to decide what to do about it. This flag is most useful if there is some type of interrupt signal that must be given priority. There is another TCP field, the urgent pointer, which indicates where the urgent data is located.
- ACK (Acknowledgement): The acknowledgement flag is used to acknowledge the receipt of data. After the three-way handshake has been completed, the acknowledgement flag is set in all TCP segments that were exchanged in the session. The receiver uses the acknowledgement number in conjunction with the acknowledgement flag to indicate the next expected TCP sequence number.

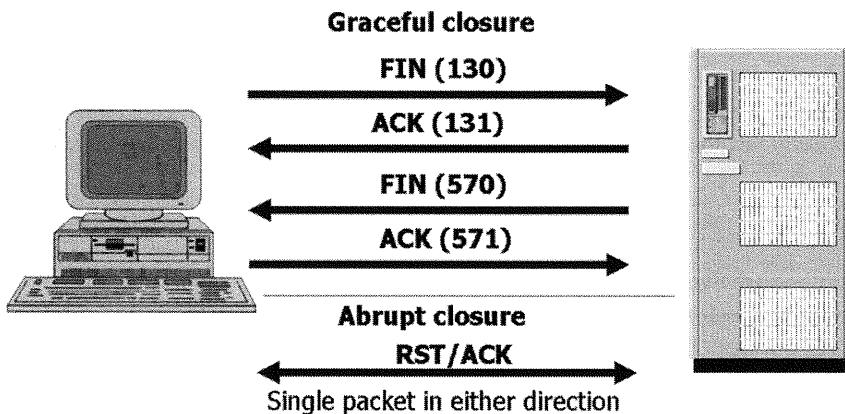
- PSH (Push): TCP stacks usually buffer incoming data until a certain amount has been collected; they then pass it in a chunk to the application. Transmitting data in bulk is usually the most efficient way to handle the stream. However, for interactive processes, such as Telnet or SSH, it is more important that data be processed as soon as it comes in, even byte-by-byte. To ask for this behavior, the sender can set the PSH flag on a packet to indicate that it should not be buffered, but instead should be passed immediately to the remote application for processing.
- RST (Reset): Immediately upon receipt of a packet with the reset flag set, a host should terminate the connection that contained that packet. Use of the reset packet is discussed briefly later in this module.
- SYN (Synchronize): The synchronize flag indicates a connection request.
- FIN (Finish): The FIN flag is the opposite of SYN. It indicates that a connection is being shut down in an orderly fashion. It contrasts with RST, in that FIN is a much more graceful way to close a connection.

TCP Checksum

The TCP checksum ensures that the TCP portion of the packet was not accidentally modified in transit. Like the other checksums we have seen so far, it is not strong enough to protect against attackers who really want to modify your packets, because they could simply compute new checksums and change this field. The checksum indicates whether or not malfunctioning routers, network congestion, or other network glitches have garbled packets.

If the receiver tries to verify the checksum and it does not match the packet it received, it simply throws the packet away and refuses to acknowledge receipt of those bytes. Eventually this will cause the sender to retransmit the packet. Hopefully the packet will come through okay the next time.

Closing a TCP Session



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Closing a TCP Session

This slide shows a sample TCP session, illustrating the two ways TCP closes connections on the network. The example assumes that a PC is connecting to a server over the network, but this same process holds true for any TCP session established between any two devices.

The arrows in the figure represent the direction of the communications. An arrow pointing from the PC to the server means that the PC is sending a message to the server; an arrow pointing from the server to the PC means that the server is sending a message to the PC. The RST, ACK, and FIN labels represent the different types of packets that are used during session setup and close. The RST packet is used to reset, or abruptly close, the communications. The ACK packet sends an acknowledgement of the message back to the originator. The FIN packet starts the process of finishing the connection. Finally, the numbers in parentheses are the sequence numbers that are sent along with each packet.

The top of the slide shows how a connection is gracefully torn down. When the time comes to close the connection, each end of the connection must be closed separately. Assuming that the PC wants to close the connection first, the process starts when the PC sends a FIN packet to the server. The FIN portion indicates to the server that the PC wants to close the connection (continuing with the sequence count it has been using with the server). The server responds by sending an ACK to the PC that is acknowledging the FIN that the PC sent. Next, the server sends a FIN packet to the PC to close its side of the connection. Finally, the PC sends an ACK to the server to acknowledge the FIN.

The bottom portion of the figure illustrates a single packet abrupt closure of a TCP session. When abruptly closing a connection between two machines, either side can send a single packet to do so.

The process starts when one system or another sends a RST packet to the other. If the receiving device is in a LISTEN state for the destination port, the RST packet is ignored. If the receiving device is in the SYN-RECEIVED state, but was previously in the LISTEN state, it returns to the LISTEN state. LISTEN state means the system is waiting for a

connection and that there are no active connections to the port or no systems trying to establish a connection. This is typically how a server operates with all open ports in a LISTEN state, waiting for a client to connect. If the receiving device has an open session and the details of the packet match with the session it references, the device goes to a CLOSED state and sends no further packets for the session and advises higher level processes that the connection has been closed. Assuming a session had already been established, the sending device will set the session to a CLOSED state. It is important to note that sessions closed in such a way are done by a single packet in either direction. No further packets are required. Closing a session in such a way is sometimes called aborting a connection.

User Datagram Protocol (UDP)

The student will understand the structure and purpose of UDP and the fields in a UDP datagram header.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

User Datagram Protocol (UDP)

This section intentionally left blank.

UDP (User Datagram Protocol)

- Connectionless communications
- Sends packets out, but does not provide any guarantee of delivery
- Much less "overhead"
- Good if small amount of packet loss is acceptable

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

UDP (User Datagram Protocol)

UDP is the simpler of the two Transport Layer protocols typically used with IP, which is why we cover it first. Even though it is simpler, UDP is a very useful, important protocol in common use by many applications today.

UDP's goal is to be a very fast, efficient protocol for reliable networks. In other words, it tries to achieve greater overall throughput by sacrificing a lot of computationally expensive error checking. Unlike some other protocols, UDP does not include the concept of a connection. The sender simply places a UDP packet on the wire without even checking to see if the receiving machine is up, let alone warning it that data is about to arrive. Furthermore, after the sender transmits a UDP packet, the sender essentially forgets about it. The sender never even confirms that the packet made it to its destination. There is also no guarantee that if the packets do arrive; they will be in the same order as they were sent. Because each packet finds its own way through the network, they often take different routes. For longer journeys, some packets inevitably will arrive out of sequence; but that is the receiving application's problem, not UDP's. The UDP header does include a simple checksum that can determine if the packet was accidentally modified en route; but technically even this is considered an optional part of the protocol (though in practice it should always be enabled). In short, UDP does very little error checking or exception handling of any kind.

It may sound as if UDP should be avoided because it does not perform error checking or have re-transmission capability. Statistically speaking, error checking is hardly ever needed on a fast, relatively error-free network, because almost all packets arrive in the proper order. By doing away with all the checking, UDP can transmit data at a much higher rate. Of course, the application then will have to assume the extra burden of planning for exceptional conditions when they occur. The error handling code will only be invoked if an actual error occurs, rather than every time a protocol operation happens. This approach saves a lot of CPU time. While the approach puts more of a burden on the application's programmer, it also provides him or her with a great deal of flexibility and power with which to work; so the tradeoff often makes sense.

UDP Uses

- The value of the packet is directly proportional to speed of delivery (Multimedia/VOIP)
- Small, single packets can carry the message (DNS)
- Multicasting is required (TCP is fundamentally incapable of multicasting.)
- Transmission is expected to occur on a reliable network. Speed is the highest priority (TFTP/NBT/NFS).
- Common UDP ports: DNS (53), Bootp (67 and 68), TFTP (69), NTP (123), NBT (137-139), SNMP (161 and 162), and NFS (2049)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

UDP Uses

UDP is typically used in situations in which it is okay if some packets are lost or reordered. In a streaming audio application, for example, each packet contains such a minuscule amount of audio data that the client probably can afford to lose one or two, packets in succession without suffering a noticeable lack of quality. In addition, because it is real-time communication, re-transmitting the packets does not make sense.

Also, UDP is often used for applications that do not send very much data, perhaps just a handful of bytes; so the applications do not mind retransmitting the data if it happens to get lost. As we saw in the last module, resolvers can query DNS servers to convert host names into IP addresses. The queries and responses usually can fit inside a single packet, so UDP is a quick and easy choice for a transport protocol. In most cases, the packets will go through fine, but the loss of one, two, or even several packets poses no great problem. The time it takes to recover from the occasional dropped packet is more than made up for by the time saved by not checking for errors that rarely happen anyway. It is easy to retransmit a query if the client does not receive a response in a reasonable amount of time.

UDP is also the protocol of choice for multicast-based applications. These applications will seek to send out the same data from one source to multiple recipients. Due to the fundamental design of TCP, this type of communication cannot occur effectively.

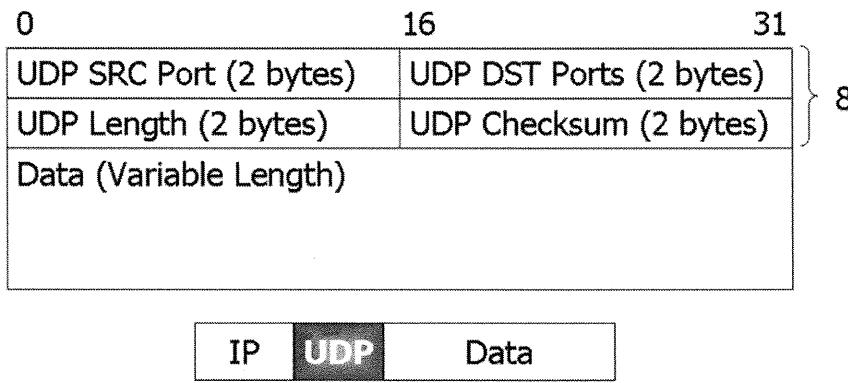
Coincidentally, the vast majority of multicast applications seen in the real world happen to fit the situation we began this page on—a multimedia application. The downside to UDP in this use is that it can congest a network due to the lack of flow control mechanisms within the protocol.

Prior to Windows 2000, the vast majority of local file transfers occurred on Windows-based networks using UDP within the NetBIOS protocols. This was a decision made to try to maximize utilization of what was expected to be a high-speed local network to take advantage of the benefits of UDP to quickly move files from one system to another. However, as networks have grown and expanded beyond local boundaries, this type of file-sharing application has begun to move to TCP due to better flow control and reliability.

Other important UDP-based protocols include:

- Network Time Protocol (NTP): Synchronizes time.
- BOOTP/DHCP protocols: Automatically configures network interfaces and load operating systems via the network when they start up.
- Network File System (NFS): Supports file sharing for Unix-based networks.
- Simple Network Management Protocol (SNMP): Used as a management tool to query network- and server-based devices for monitoring or troubleshooting purposes.
- Trivial File Transfer Protocol (TFTP): Used as a method to transfer files from one device to another without requiring authentication. TFTP's most common use is in updating code on network-based devices.

UDP Header



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC.

UDP Header

As packet headers go, UDP is pretty simple. There are only four fields: source port, destination port, datagram length, and checksum. Each field is exactly two bytes long. A mere 8 bytes of overhead per packet is pretty good! Let's examine these fields in detail.

Source Port & Destination Port

UDP uses the concept of ports to help get datagrams to and from the proper applications. Ports are simply ID numbers associated with certain applications running on a host. When one host wants to send datagrams to a server process running on another host, it needs to know what port that process is listening to. If we consider a computer to be similar to an apartment building, the applications running on it are its residents and the port numbers are like the apartment numbers in which the residents live. DNS DuBois lives in Apartment 53, for example, so messages (packets) going to that apartment number clearly are meant for him.

Most server ports are well-known, like DNS servers that always listen to port 53 no matter how many times they are restarted or the machine is rebooted. Well-known ports are usually considered those from 1 to 1023. Clients usually use ephemeral ports -- ports that change each time the client application runs. Ephemeral ports are numbered above the well-known ports (greater than 1023).

For TCP and UDP, the source port indicates the port to which the sender is bound, while the destination port indicates the service on the receiver to which the packet should be delivered. Valid port numbers are 0 through 65,535.

UDP Datagram Length

Datagram length is simply the length of the UDP portion of the packet, which includes the UDP header as well as the payload. Theoretically a datagram could carry no data, setting the minimum value here at 8 (just the size of the header). The theoretical maximum is 65,535, though many implementations do not allow datagrams to be that long.

Checksum

The datagram's checksum is technically an optional component, though almost every UDP implementation uses it. If specified, it allows the Transport Layer to detect when the UDP headers or the payload data (but not the IP headers, which have their own checksum) have been modified in transit. This is trivial to recompute, so an attacker interested in modifying a UDP packet will have no problem doing so and then generating a new checksum. This really isn't a security feature so much as a way to detect accidental transmission problems.

UDP Summary

UDP is a great choice for a transport protocol if you have a fast, reliable network and need either high throughput or quick response times (or both). By avoiding expensive error checking, applications can take advantage of UDP's quick and responsive nature. Still, it is not a perfect protocol for all uses. Its greatest strength is also one of its greatest weaknesses. Because it does no error checking of its own, the application programmer must take up this burden. On many networks, especially WANs or the Internet, packets routinely are lost or mangled. A more robust protocol that can handle these situations automatically would be more desirable. This is, in fact, the whole reason for the existence of TCP.

TCP and UDP

- Transport Control Protocol:
 - Guaranteed delivery
 - Connection-oriented
 - Additional protocol overhead (slower)
 - Optimized for sessions
- User Datagram Protocol:
 - Delivery not guaranteed
 - Connectionless
 - Less protocol overhead (faster)
 - Optimized for query responses

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

TCP and UDP

We hope by now you have a much better idea of how much work TCP puts in to making sure your connections are as efficient and as error-free as possible. Having all that built-in error correction capability means a tradeoff between raw speed and reliable communications, but for most applications, especially those designed for the Internet, it is probably well worth it. Reliability is easier to program for because TCP takes care of a lot of the details for you. UDP, on the other hand, is good for real-time communication, where guaranteed delivery offers little benefit.

TCP vs. UDP Ports

It is worthwhile to point out that although UDP and TCP port numbers look similar, they represent different protocols and do not overlap at all: the two protocols keep their port numbers separate. It is quite possible for a TCP application to listen on TCP port 107, for example, while an entirely different application listens to UDP port 107.

That being said, it is also common for a service to bind to the same TCP and UDP port numbers for both connections, just to ensure that no matter which protocol a client chooses to use, the server will receive the information. This is not a requirement, nor does it happen automatically, the application has to be written this way. TCP and UDP port numbers may look the same, but they are in separate address spaces.

Internet Control Message Protocol (ICMP)

The student will understand the structure and purpose of ICMP and the fields in an ICMP datagram header.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Internet Control Message Protocol (ICMP)

This section intentionally left blank.

ICMP (Internet Control Message Protocol)

- 2 purposes:
 - To report errors or troubleshooting
 - Destination host unreachable
 - Fragmentation needed and DF flag set
 - To provide network information
 - Ping: Is the host alive and what's the latency?
- Tied to version of IP:
 - ICMPv6 is implemented for IPv6 networks

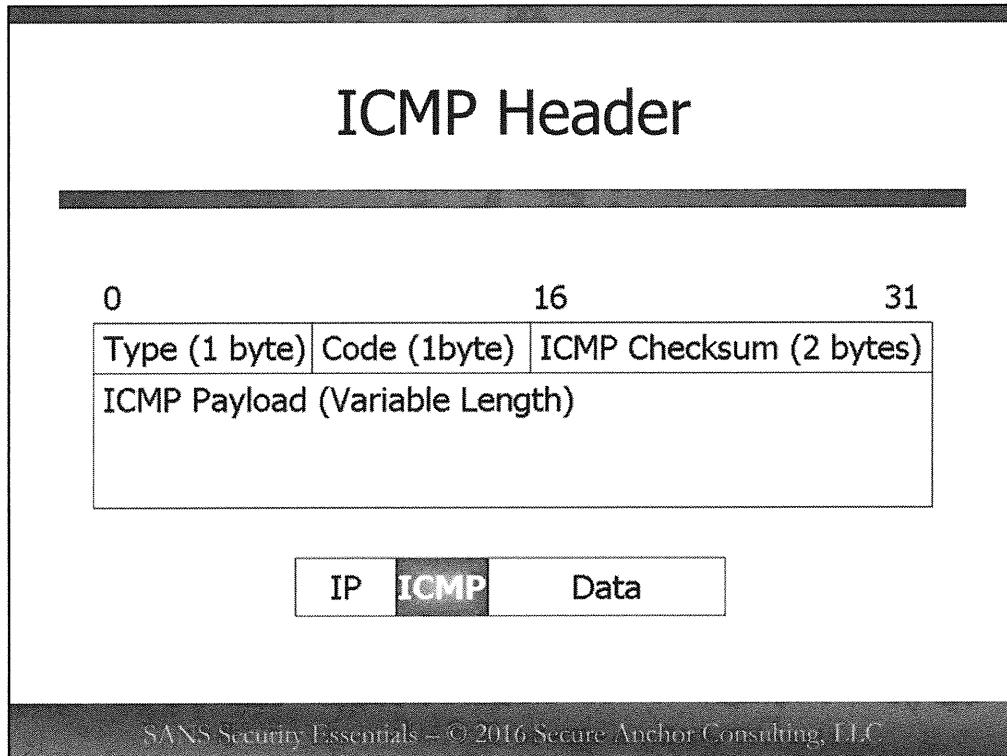
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

ICMP (Internet Control Message Protocol)

The third (and final) protocol we discuss in this module is the Internet Control Message Protocol (ICMP). We mentioned this briefly in the last module, but this is a very good time to cover it in more detail. ICMP is a Network Layer protocol, unlike TCP and UDP, which are part of the Transport Layer. As such, ICMP actually is a peer of IP, even though it is still encapsulated in an IP packet. In fact, IP, TCP, and UDP all rely on ICMP to provide information about network conditions, as well as for status and error messages pertaining to their transmissions.

ICMP is a simple protocol. It is datagram-based, like IP and UDP. Most ICMP transactions require only one or two packets. ICMP packets have only three header fields, fewer fields than UDP, and one of them is just a checksum!

ICMP is an important protocol because it carries critical and fundamental information about the state of a network and error conditions that occur. It is not meant as a protocol to be used for the transmission of data; rather, it is designed for error reporting and network-based troubleshooting techniques. Many of the other protocols rely upon ICMP to perform functions and communicate error conditions.



ICMP Header

The ICMP header, like the TCP header discussed earlier in this module, includes various types of fields. Each field is described here.

ICMP Type

The type field contains an integer that identifies which type of ICMP packet is being sent. There are 8 bits allocated to hold the type. Check out the IANA page (at <http://www.iana.org/assignments/icmp-parameters>), which lists the many defined options for the ICMP type field.

ICMP Code

The code also has a bearing on the type. For many messages, ICMP code acts as a sort of sub-type. When the type field is 3, the packet is an ICMP Destination Unreachable packet. The code can give the receiver much more detailed information. A code of 3 indicates that the host is available, but the specific port requested is not listening. A code of 9 might indicate that a router or firewall rule blocked your communication to the remote host.

ICMP Checksum

The ICMP checksum is computed as a 16-bit one's complement of the header and data portion of an ICMP packet, assuming the checksum field itself is set to all zeroes.

The ICMP Payload

The content of the packet's payload might also be important to the receiver. When a host generates an ICMP error message, it always includes the entire IP header of the packet that caused the error condition. It also includes the first 8 bytes of the IP payload, which is the beginning of the TCP or UDP header containing the source and destination ports. This lets the original sender know exactly which packet caused the error and, consequently, to which application it should deliver the error message.

Common Types and Codes

- Type 0: Echo reply
- Type 3: Destination unreachable
 - Code 0: Network unreachable
 - Code 1: Host unreachable
 - Code 3: Port unreachable
 - Code 9: Destination network administratively prohibited
- Type 5 - Redirect
- Type 8 - Echo request
- Type 11 - Time exceeded
 - Code 0: TTL expired in transit
 - Code 1: TTL expired during reassembly

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Common Types and Codes

The types and codes depicted on the slides cover only a small portion of all the types and codes that exist. However, these are a good representative sample of the most common types of ICMP packets that we need to familiarize ourselves with.

Type 0 - Echo reply

A Type 0 packet currently has no codes that carry any special meaning. This packet is typically referred to as a ping response to an ICMP Type 8 packet. The Type 0 packet is used to tell us that the remote host is reachable on the network and the amount of latency (measured by the sending host) to that device.

Type 3 - Destination Unreachable

At this time, there are a total of 15 codes associated with this ICMP type. There can be many reasons why a destination may be unreachable. The code value is meant to provide a little bit more information as to the reason that the remote host cannot be reached. If a router is unable to pass a packet to the next hop for a destination network (such as if an ISP connection goes down), it is probable that a "network unreachable" code will be returned to the sending host. However, if the ISP connection is up but the destination host simply doesn't exist on the network, then a "host unreachable" code will be returned. Further, if the host does exist on the network, but the requested UDP port (usually only seen used for UDP, as TCP has its own mechanism for handling this) is not an open port, then a "port unreachable" code may be returned.

Finally, if some type of administrative access control list prevents access to the destination network, even if the network path fundamentally exists, a "destination network administratively prohibited" code may be returned. These are just a few of the codes associated with this type.

Type 5 - Redirect

There are currently four codes associated with a Type 5 ICMP packet. This type is used to affect the routing table on the receiving device to change where packets are sent. The Type 5 packet must be used cautiously because an attacker can potentially abuse these types of ICMP packets to redirect traffic to a location where he or she can easily view and manipulate them. In many cases, routers are configured to ignore ICMP redirect packets for just this reason.

Type 8 - Echo Request

A Type 8 packet is used to elicit a response. Preferably, the response will be an ICMP Type 0 packet, but due to any number of issues, the response could really be a variety of other ICMP type/codes depending upon the status of the network. Most of the time, when people refer to ICMP packets, they are referring to them in the context of the ping protocol that works mostly with Type 8 and Type 0 traffic.

Type 11 - Time exceeded

Type 11 traffic currently has two codes associated with it. The most common reason for an ICMP Type 11 packet to be sent is that the time to live (TTL) value has been exceeded. This may occur in the event of a routing loop or if the initial TTL value was set too low to begin with. It's also used in some cases as a method to perform a traceroute, which will be discussed later in this module. One code signals this occurrence. The other code covers timeouts that occur due to fragment reassembly time being exceeded.

Ping

- Determine whether a destination host is active
- Determine the latency between two locations
- Determine the rate of packet loss
- There are some security concerns
- Some sites block ICMP:
 - Covert data channel
 - Denial of Service attack
 - Used to map a network

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Ping

One of the most common uses for ICMP is to test whether a host is up and accepting network connections. ICMP defines the Echo Request (Type 8) and Echo Response (Type 0) messages to provide this information. The usual way to send these packets is to use the ping command. Although some of the details of the ping command differ among operating systems, the basics are always the same. Ping sends several ICMP Echo Request packets to the remote machine and waits for their replies. When it receives replies, it prints out a few statistics about them. The most useful statistic is probably the round trip ping time. The ping command keeps track of when it sends the Echo Request packet and when it receives the corresponding Echo Reply. The difference between those two times is the round trip time. In other words, it gives an indication of how long it took both the request and the reply packets to travel the network links between the two computers, which should tell you something about the latency of the end to end connection. Anything under 10 milliseconds is probably your local LAN, although ping times of 200 or 300 ms (or more) are not uncommon over a WAN like the Internet.

Traceroute

The student will be able to use the traceroute utility for network troubleshooting and discovery.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Traceroute

This section intentionally left blank.

Traceroute

- Traceroute shows you the path a packet takes to reach its destination
- It can tell you a route's external router and be used to map a network
- A normal traceroute lists the routers
- General rule: All hosts on the same network must go through the same external router and, potentially, the same firewall
- By performing traceroutes and looking at the last couple of hops, you can map out a network

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Traceroute

Traceroute uses a clever combination of TTL values and ICMP replies to map out the route packets take from one computer to another, sometimes through many hops. The command works by sending a series of packets all going to the same destination, with TTL values starting at 1. When the first packet is sent, its TTL expires at the first hop, so the router usually replies with an ICMP "Destination Unreachable" or "Time Exceeded" message. The traceroute command eventually receives this reply and looks inside its payload for the IP address of the sender, which it assumes is the first hop's router.

Traceroute then sends a second packet, this time with a TTL of 2, which expires at the second hop, generating another ICMP reply. Traceroute now knows the second hop's router as well. It keeps sending packets this way, incrementing the TTL by 1 each time and getting replies from each hop until one of the packets finally is delivered to the destination host. By continually incrementing the TTL, traceroute can record all the routers in the path the packets took between your machine and some other machine on the Internet.

Summary

- **TCP:** Guaranteed delivery, connection-oriented, additional overhead
 - Three-way handshake, positive acknowledgement
- **UDP:** Fast with little overhead, connectionless, and no guaranteed delivery
- **ICMP:** Error reporting and troubleshooting
- **Traceroute:** Determine the network path taken to a destination host

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

We have crammed a lot of information about TCP, UDP, ICMP, and traceroute into this one short module. The contrast between UDP and TCP (quick and efficient versus more overhead but more fault-tolerant) means that the two protocols are radically different, even though they are both based on the same underlying Network Layer, IP.

One of the most obvious differences between UDP and TCP is the size of their packet headers. TCP, being a more detailed protocol, requires a lot more information about a given packet. UDP gets by with a mere 8 bytes of header information, while TCP requires at least 20. The two protocols have some fields in common, such as the source and destination ports, but TCP adds several other fields designed to help facilitate efficient, reliable transmission.

We also talked a lot about TCP connections: what they are, how to establish them, and how to close them down. A single TCP connection actually involves two channels, one for sending data and another for receiving it. Closing a connection requires a similar procedure or a reset. Knowledge of both of these processes undoubtedly will be useful to you when you're trying to analyze suspicious behavior on your network.

You also should have a pretty good understanding of ICMP's role in an IP network by now. ICMP is a Network Layer protocol just like IP, but it is not typically used to send application data. Instead, IP uses ICMP to convey information about hosts and network conditions. Two of the most common uses for ICMP include the ping program, which tests to see whether a host is responding on the network, and the traceroute command, which maps the route packets take as they travel over the network. ICMP also signals certain error conditions to an IP stack. For example, if a router is unable to deliver a packet, it will usually return some sort of "Destination Unreachable" message to the sender. You can think of ICMP as the signaling protocol used to let IP stacks communicate conditions with each other.

Finally, you should have an understanding of how the traceroute command typically uses ICMP packets in a Windows environment (with the tracert.exe command) and UDP traffic in the Unix environment in conjunction with carefully managed Time to Live (TTL) values in order to determine the network path between two locations. The information available from traceroute is valuable in assisting with mapping a network, determining potential routers and firewalls, and identifying potential problems areas on the network where latency is dramatically increased.

Module 5: Protocol Analysis

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 5: Protocol Analysis

This section intentionally left blank.

Protocol Analysis

SANS Security Essentials I: Networking Concepts

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Introduction - Protocol Analysis

If you do not know how something works, you cannot troubleshoot it. The same is true for networks. If you do not understand what you are looking at on a network, how can you tell what is normal and what is erratic? What is authorized and what is malicious? You cannot determine the answers if you do not understand what you are looking for on the network.

Objectives

- Network sniffing
- Hubs versus switches
- Sniffers
- Analyzing packets

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

This module covers an invaluable network auditing tool—the sniffer. We show you how to capture network data, analyze it, and look for patterns. The sniffer allows network administrators to visualize and analyze the data on their networks. We look at one particular free sniffer, tcpdump, in greater depth, and provide real-world examples of how it can be used. This module demonstrates how network analysis often results in understanding the inner workings of various IP protocols. Because of its complexity and widespread use, we give one particular protocol—the Transmission Control Protocol (TCP)—special attention.

The ability to capture network data is essential to network-based intrusion detection systems (NIDS), which gather data from sensors on a network, correlate it, and perform automated analyses on the data. When a common attack pattern is recognized, such as a piece of exploit code or a known password for a distributed denial-of-service zombie, the NIDS alerts the security response team. This team has to be able to look at the data that caused the alert and determine whether it is innocuous (a false positive) or if it warrants a response.

Knowing the normal usage patterns on your network will help you recognize something abnormal when it happens. Seeing the reconnaissance probes and exploits that are attempted against your network continually is also useful, giving you a better sense of your threat model and a better understanding of what you have to prepare for. When it comes to network security, knowledge is power; the more you know about your network traffic, the better.

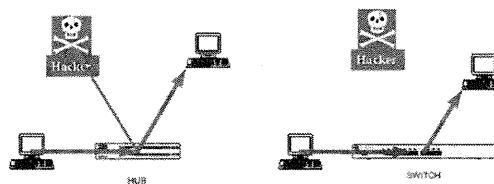
Some network administrators are surprised to find out how much malicious traffic impinges on their networks every day. Internet-connected systems are almost guaranteed to be probed several times per day.

Network traffic itself provides valuable information. You can even use a sniffer to prove that a certain kind of traffic is blocked or passes through your network without interruption, if a service is ready and available for your users, and how much broadcast traffic crosses your LAN boundaries. Looking at your network traffic, you may be surprised to learn how many protocols and services are offered or sought by network nodes. Looking at these network traces helps you build a baseline of what is normal inside your network and helps you to investigate abnormal traffic patterns.

The proper analysis skills are invaluable in an emergency, be it an attack or a catastrophic failure. When documentation and technical support fail you, often you can count on a packet dump—a sample of network data—to tell you what is really going on.

What is a Sniffer?

- A sniffer is a program and/or device that monitors data traveling over a network
- Sniffers can be used for legitimate network functions. Unauthorized sniffers can be extremely dangerous to a network's security
 - Broadcast media (Ethernet) allows an attacker to steal information off the network; for example, an attacker might gather passwords
 - For Ethernet, all data is broadcast on the LAN segment
 - The hub sends data to every system
 - A switch limits data to a specific source and destination port on a switch



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

What is a Sniffer?

Data is sent across a network in discrete bundles called packets. For a sniffer to see packets, they have to arrive on the network interface of the host running the sniffer. Out of all the packets traversing the network, which ones reach the sniffer host depends on the network device connecting the hosts on the network.

Broadcast Versus Switched Ethernet

One of the most popular networking technologies is Ethernet. Each device on an Ethernet, be it a computer, router, or hardware sniffer, has a 6-byte physical address or Ethernet address. This physical or Ethernet address is the same as the MAC address. Devices with multiple Ethernet interfaces often have different physical addresses for each interface, but not always. Usually the physical address is written as a series of hexadecimal numbers, each representing a byte, separated by colons (00:30:65:01:b8:c5, for example). The physical address of a device has nothing to do with its IP address. Devices on an Ethernet use the Address Resolution Protocol (ARP) to determine what physical address an IP is associated with, so they know where to send IP datagrams.

An Ethernet packet, also called a frame, contains in its header the physical addresses of the source and destination devices. Usually a device automatically accepts a frame whose destination address is the same as its own.

Broadcast or Shared Networking

Hubs connect traditional Ethernet networks. When a hub sees an incoming frame, it simply broadcasts it out all of its ports. Such shared Ethernets are inherently sniffable; all of the data transmitted by any host is visible to all other hosts on the hub.

Because of the broadcast nature of shared Ethernet, hosts have to be selective about which frames they accept. A host normally rejects frames that are not destined for it. But to get as large a sample as possible, sniffers use the interface's promiscuous mode to accept all frames, even those intended for other hosts. It's not a bad idea to check network

interfaces periodically to see if they are in promiscuous mode without your knowledge. If they are, an attacker might have compromised your machine and installed a sniffer. To check your interface in Unix, use the ifconfig command and look for the string PROMISC. Windows does not build in this functionality.

Switched Networking

The Ethernet switch is more intelligent than the hub. By inspecting incoming frames, the switch determines the physical address of the destination host and then finds the port on the switch to which the system is connected. Then it sends the packet out on the correct port.

Because a sniffer on a switched network sees only traffic to and from the sniffer's host, promiscuous mode has no effect. For this reason, the use of switched networking is considered a security feature. After all, a compromised machine on a switched network can sniff only passwords and sensitive data going to and from that machine, something that could be done by other means anyway (keyboard sniffing or just perusing the file system).

But it is still possible to sniff traffic from other hosts on a switched Ethernet network by impersonating a local router, for example. Until recently, this technique was widely deemed impractical, but newer tools such as dsniff and ettercap facilitate sniffing in a switched environment. Even though switched networking does improve security by preventing casual sniffing, strong cryptography should be used to protect passwords and sensitive data.

A packet sniffer can be used to:

- Monitor network usage
- Gather and report network statistics
- Analyze network problems
- Debug client/server communications
- Detect network intrusion attempts
- Gain information for effecting a network intrusion
- Filter suspect content from network traffic
- Spy on other network users and collect sensitive information such as passwords
- Reverse engineer protocols used over the network

Examples of Sniffers

- There are countless examples of sniffers, such as the following:
- tcpdump: Freeware
- windump: Freeware
- Wireshark: Freeware
- Snort: Freeware
- Es: Freeware (ships with SunOS, Solaris Rootkits)
- Linsniff: Freeware (ships with Linux Rootkits)
- Websniff: Freeware Snoop (distributed with Solaris)
- Network General: Commercial
- Sniffit: Freeware
- Dsniff: Free suite of tools built around a sniffer
- Kismet: Free 802.11 layer 2 wireless network sniffer and intrusion detection system

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Examples of Sniffers

The first step in analyzing probes and other malicious traffic is to see it with your own eyes. It is common sense to want to see the guts of a system when it is not working properly. That is why the sniffer is a favorite tool of system and network administrators. Because sniffers can gather all information transmitted on a network at a given time, including passwords and other sensitive data, they are also a favorite among attackers.

Sniffers can be hardware devices that physically attach to the network, but more commonly they are software programs that run on networked computers. The sniffers that come bundled with your operating system are designed as tools for the system administrator. No matter what your needs, your interest, or your budget, there probably is a sniffer out there that does what you want. Some sniffers are designed for more specialized, nefarious purposes. Sniffers bundled with rootkits often are designed to seek out usernames and passwords in the network data and extract them into files. To use this type of sniffer, the attackers do not need any technical knowledge. They just run a program, and after a while, they have a file full of usernames and passwords they can use for further intrusion on the network.

Attackers can also use your own sniffers against you. Is this an argument against using sniffers as network analysis tools? Of course not; sniffers are too valuable for you to do without entirely, and attackers can always bring their own. But it's worth the effort to keep sniffers out of easy reach of a potential attacker. It's bad enough if one of your production servers gets compromised, but worse still if the attacker is able to use a locally installed copy of tcpdump to capture passwords or other sensitive information. It's a good idea to keep your sniffer software where it can be controlled tightly, perhaps on a laptop specifically designated for network auditing.

Sniffing on a Switch

- Sniffing traffic allows an unauthorized computer user to view the traffic destined for someone else
- Unlike hubs, switches prevent promiscuous sniffing
- Traditional unauthorized sniffing on a switch is difficult, but with the advent of tools like dsniff, it has simplified this task. With an ARP redirect program and IP forwarding, an attacker can sniff every station on your switched network.
- Most switches support "port mirroring," SPAN, "management port," or similar features, which allow network administrators to perform authorized sniffing to monitor LAN traffic on any computer connected to one designated switch port

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Sniffing on a Switch

To sniff traffic on a switched segment is not an impossible operation. Dsniff is software that accomplishes this task in a simple way: The attacker's system sends out a forged ARP packet to the target system telling it that its default gateway has changed to the attacker's system. When the target system sends traffic on the network, it will send it to the attacker's system first, which then forwards the packet on to its original destination as if nothing ever happened.

Of course, the attacker will need to use either the kernel-level IP forwarding in /proc/sys/net/ipv4/ip_forward or fragrouter on a Linux system to perform the packet forwarding. So, by forging ARP replies for the default gateway of a network, all traffic destined for the default gateway will be sent to and then forwarded by the attack system. Once received at your system, you can grab anything you desire, including passwords such as SNMP, FTP, POP (post office protocol), HTTP, IRC (Internet Relay Chat), Telnet, and many others.

Other utilities that can be used to perform this task include:

- Mailsnarf
- Webspy
- Ettercap

Ettercap is a second generation sniffer that allows you to sniff all the traffic, even in a switched network. Ettercap relies heavily on ARP cache poisoning (the attacker sends false ARP replies to associate his MAC address to the IP addresses of both the source and the target hosts).

tcpdump/windump

The student will be able to use the tcpdump or windump utility to read packets from a network interface and understand the output.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

tcpdump/windump

This section intentionally left blank.

tcpdump

- tcpdump is a program that dumps traffic on a network and is dependent on the libpcap packet capture library
- tcpdump has also been ported to Windows as windump and is dependent on winpcap
- tcpdump is a tool that is universally used and very portable
- tcpdump is a sniffer and does not attempt to make interpretations of what it sees; it is not a protocol analysis tool

The screenshot shows a terminal window titled 'root@kali: ~'. The command 'tcpdump -i mon0' is being run. The output displays several network packets in ASCII and hex dump format, illustrating the raw data captured by the sniffer.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

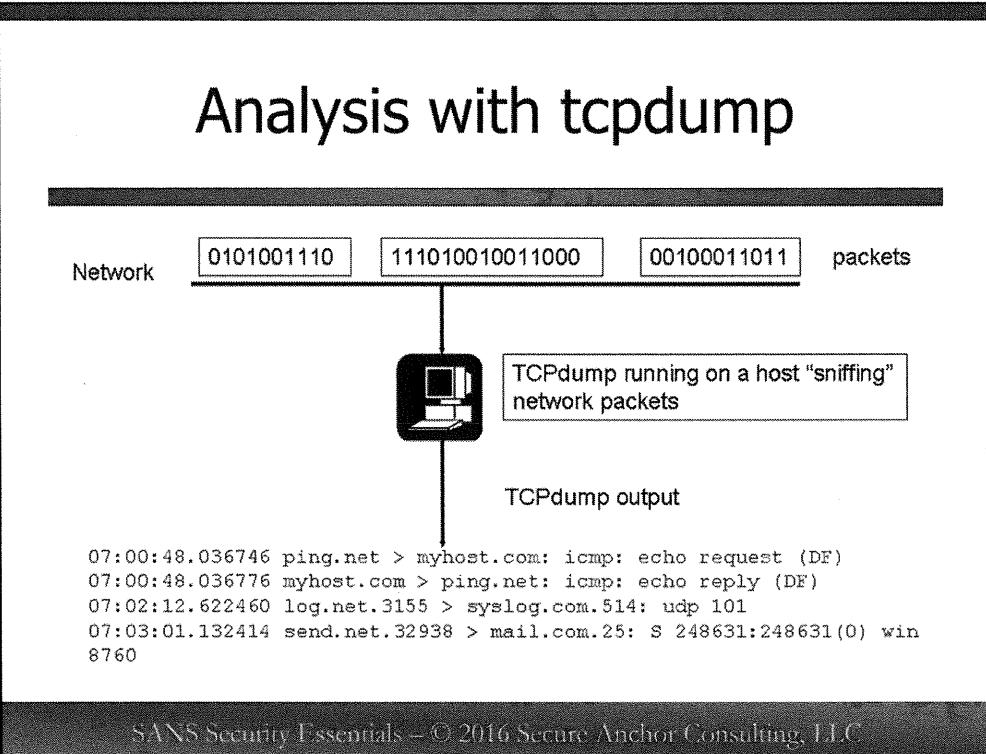
tcpdump

One of the most popular sniffers available is tcpdump, which runs primarily on Unix and even ships with a few Unix distributions. A Windows port, windump, is also available. Both are free.

This section covers protocol analysis with tcpdump. Its command-line interface is not fancy, but does provide a versatile filtering language. Command-line options can be used to control tcpdump's behavior. After an overview of filters and command-line options, we move on to actual tcpdump analysis of UDP, TCP, and ICMP traffic. Because each protocol behaves differently on the network, tcpdump uses a different format for displaying them.

tcpdump is a tool that is universally used and very portable. It can be used on just about any platform to assist you in the analysis of network traffic. It does not attempt to make interpretations of what it sees; therefore, you, the analyst, have to have the training and knowledge to interpret the data. For the sophisticated analyst, this is a bonus because she or he can make the correct call. For an analyst who has little training, tcpdump can be daunting because it does not interpret events.

Analysis with tcpdump



Analysis with tcpdump

By default, tcpdump collects all the packets visible to the host it runs on and produces formatted output containing information on those packets. On a busy network, tcpdump with no arguments will generate output faster than you can read it, let alone analyze it. You probably would not be interested in most of the output anyway.

You can cut the output down to a more manageable size through the use of filters - Boolean expressions evaluated for each packet tcpdump sees. Filters can be specified right on the command line or in a file specified by the -F option.

If you are only looking for Telnet traffic, it would be tedious to wade through all the TCP traffic on the network and pick out just the packets destined for port 23. So you let the filter do the work for you. The word dst port specifies that you will only see packets headed for that port, so you will not see any return traffic from the telnet server back to the client. Note that just because a packet is heading for port 23 does not mean that it is Telnet traffic; it is simply a good guess. You would have to dump the raw packet, which you will learn how to do later, and look for evidence of the Telnet protocol in the TCP payload.

tcpdump tcp	only dump TCP packets
tcpdump tcp and dst port 23	only dump TCP packets destined for port 23 (usually used for telnet)
tcpdump host nmap.edu	only dump packets to or from nmap.edu

Often you are interested in all traffic involving a particular machine. In the third filter, you tell tcpdump to show only traffic to or from nmap.edu. Without the dst keyword, tcpdump returns packets with nmap.edu in either the source or the destination fields of the IP datagram.

Almost any field, including the actual data payload, can be used to select the packets that are collected. Plus, a full set of logical operators are available (notice the "and" in the second example) for joining an arbitrary number of clauses, allowing you to be restrictive enough that you do not have to look at screens of output.

Command-Line Options

Several of tcpdump's command-line options are worth mentioning. Because tcpdump often is used to diagnose network problems, it needs to work well when certain core network services, such as DNS, are not available. By default, tcpdump will try to resolve all IP addresses to names. If DNS is down or malfunctioning, those reverse lookups can take a long time, and you must wait awhile before you see each packet come in. The -n disables these lookups, so it is a recommended option when DNS is unreliable or during the initial troubleshooting phase. Because -n avoids delays caused by some slow name servers out on the Internet and ensures the immediate display of each packet, some people use this option all the time, whether or not DNS is failing.

tcpdump is not just a sniffer; it is also a simple protocol analyzer that knows how to interpret some of the application data and present it to you. DNS is one example; tcpdump tells you whether a DNS packet is a query to resolve an A record, a name server response, and so on. tcpdump only goes so far in detailing this information for you, but you can request more information with -v, still more with -vv, and the most possible information with -vvv.

Contrary to what you might expect, tcpdump does not necessarily read the entire packet. So regardless of how much verbosity you request, tcpdump can only show you details for the portion of the packet it has sniffed. By default, it only reads the first 68 bytes. You can change this by specifying -s (which stands for "snap length") followed by the number of bytes to read in from each packet. Note that the 68-byte default and the argument to -s includes all header bytes, not only the payload. It is usually a good idea to use -s 1500 if you want to capture the entire packet because Ethernet has a maximum segment size of 1500 bytes. In later versions of tcpdump, -s 0 will read in the whole packet, regardless of its size.

This slide depicts tcpdump in action, collecting packets from the network interface and producing formatted output as shown. tcpdump's output has a default format based on the protocol (such as TCP, UDP, or ICMP) of the packet that is displayed.

This slide shows the following records:

07:00:48.036746 ping.net > myhost.com: icmp: echo request (DF)
This is an ICMP echo request packet sent by host ping.net to myhost.com.

07:00:48.036776 myhost.com > ping.net: icmp: echo reply (DF)
This record shows the reply from myhost.com to ping.net: it is an ICMP echo reply packet.

07:02:12.622460 log.net.3155 > syslog.com.514: udp 101
This record shows a syslog entry sent from host log.net to host syslog.com using the UDP protocol with destination port 514.

07:03:01.132414 send.net.32938 > mail.com.25: S 248631:248631(0) win 8760
Finally, this is a typical start of a TCP session, the first stage of the so-called three-way handshake and marks the beginning of an e-mail transaction between the host send.net and what we presume to be the mail server, mail.com. Note the destination port 25.

Sample tcpdump ICMP Output

ICMP format 1

```
timestamp source dest icmp: icmp message  
14:59:30.220000 ping.net > hosta.mysite.com: icmp: echo request  
14:59:38.140000 hosta.mysite.com > ping.net: icmp: echo reply
```

ICMP format 2

```
timestamp router source icmp: dest icmp message  
02:09:47.600000 foreign.router > tryinghost.com: icmp: host  
desired.com
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Sample tcpdump ICMP Output

ICMP is used in conjunction with other Internet protocols for out-of-band messaging for error handling and diagnostics. ICMP does not use ports like TCP and UDP; instead, it has a fixed number of message types and codes understood by all devices.

ICMP is also heavily used for UDP transactions because of their lack of state. Whereas a TCP packet with the RST flag set terminates a rejected TCP connection, an unexpected UDP packet must be rebuked out-of-band with an ICMP port unreachable packet.

This slide shows an ICMP message sent from ping.net to hosta.mysite.com, followed by a response. Because ICMP does not use ports, tcpdump shows us only the hostnames, tells us it's an ICMP packet, and then displays the message type. The first record is for an ICMP echo request as generated by the famous ping program. The response to the ping packet is an ICMP echo reply. Most likely, these records show normal activity - someone on ping.net using ping to check the availability of hosta.mysite.com.

A router is often involved when some kind of error is detected. In the record shown in the bottom of the slide, a foreign.router delivers a message to tryinghost.com that host desired.com is not reachable. This implies that tryinghost.com first attempted to send some kind of traffic to desired.com, but foreign.router intervened to inform tryinghost.com after it discovered a problem.

Sample tcpdump UDP Output

timestamp source.port dest.port : udp bytes

09:39:19.470000 nmap.edu.728 > dns.net.111: udp 56

timestamp: Measured as hour, minutes, second, fractions of seconds

source.port: Source IP/hostname.source port

dest.port: Destination IP/host.destination port

udp: May or may not expressly label the UDP protocol

bytes: Number of bytes of UDP data (payload)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Sample tcpdump UDP Output

UDP is an efficient, no-frills protocol. As noted in the previous module, it offers no guarantees that datagrams arrive in order or without errors. As a result, an application must either not care too much that the data arrives intact or it must provide its own reliability mechanism. Most of the applications that use UDP do so for efficiency or to avoid network congestion. DNS, NFS, and syslog are some common examples of UDP services.

This slide describes some common fields in a UDP record. The timestamp field contains the time of day tcpdump captured the packet, to microsecond precision. See how the timestamp ends in four zeros? This particular example was collected on a system running a version of the Linux kernel that did not yet support microsecond precision on record timestamps.

The source.port field contains the source host and port separated by a dot (.). The host is represented as a DNS hostname (nmap.edu) or an IP address if it cannot be resolved via DNS. Use the -n option to prevent tcpdump from resolving any IP addresses. The source port number in this example is 728.

Similarly, the dest.port field contains the destination host (dns.net) and port (111). UDP port 111 often is used for the Sun Remote Procedure Call (RPC) portmapper service and is referred to as portmap or sunrpc.

The field labeled udp contains the string, udp.

The bytes field contains the number of bytes in the payload of the UDP packet. Because UDP is a Transport Layer protocol that runs on top of IP, the UDP packet is encapsulated with an IP header before it is sent out on the network. The bytes field does not account for those header lengths—only the length of the UDP payload.

Sample tcpdump TCP Output

timestamp	source.port	dest.port	flags	beginning seq #	ending seq #	bytes	options
09:32:43.910000	nmap.edu.1173	> dns.net.21:	S	62697789	:62697789(0)	win 512	

flags: tcp flags (PSH, RST, SYN, FIN)
beginning seq #: for the initial connection, this is the initial sequence number (ISN) from the source IP
ending seq #: this is the beginning sequence number + data bytes
bytes: data bytes (payload) in the tcp packet
options: options that the source host advertises to the destination host

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Sample tcpdump TCP Output

Like UDP, TCP is a Transport Layer protocol. The similarity ends there, however. At the cost of higher overhead, TCP provides a reliable stream for exchanging data. HTTP, FTP, telnet, and many more Internet services use TCP.

Despite the differences in TCP and UDP packets, their corresponding tcpdump records share several fields in common. In fact, the records are identical as far as the timestamp field and the source and destination host and port fields. What distinguishes the TCP format from the others are the fields representing TCP flags, sequence numbers, acknowledgment numbers and options.

TCP packets often are called segments. The flags in a TCP segment represent certain qualities of the segment, often having to do with the current state of the connection. Some of the more important flags are listed, in their often abbreviated form, in Table: TCP flags. You should notice in the slide that the SYN flag is set. tcpdump abbreviates the flags further, only displaying the first letter. More than one flag at a time might be set, but only certain combinations are valid for TCP. When no flags are set, tcpdump will display a single dot (.) as a placeholder.

One of the ways TCP guarantees reliable packet delivery is by keeping track of the data it has received. This is done using sequence numbers. The slide shows the initial packet of the connection; its beginning sequence number also is known as the initial sequence number (ISN). The ending sequence number is then just the sum of the ISN plus the number of data bytes sent in this TCP segment. The beginning sequence number in the slides is 62697789. Because the connection begins with this packet, that number is also the ISN. Just after the colon (:) is the ending sequence number, 62697789.

As in the UDP record, the bytes field represents the number of bytes in the payload. As part of the three-way handshake, a segment with the SYN flag set sends no data bytes, as represented by the zero in parentheses. Data should not be sent until the client and server actually complete establishing the connection.

Each host uses the options field to advertise certain characteristics about itself or its local network to make data exchange efficient. In this record, nmap.edu advertises a window size, or an incoming buffer size, of 512 bytes to the remote host. If dns.net has a faster CPU or more memory, it might have to pace itself when sending data, so it does not exhaust resources on nmap.edu.

Reading Packets

Manually Inspecting Packet Fields for Analysis

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Reading Packets

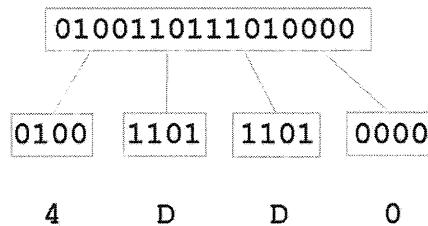
The next several slides are designed to show you how to "read" a packet header, using an IP packet with a TCP payload as an example.

Being able to manually read packet headers is not the easiest task, and some people may question whether the information is very valuable when modern packet sniffers can decode packets for you. We at SANS want to make sure you are best equipped to succeed in an information security role, and we know from experience that there will be times that you will have to manually evaluate the contents of packets to identify what they are doing. For instance, there have been many cases where an attacker intentionally puts incorrect information in packet headers to evade detection by an intrusion detection system, or to exploit vulnerability in how an operating system extracts the data from packets. If you rely on a packet sniffer to decode the contents of fields, you might get incorrect or inadequate results that would make it difficult to identify malicious behavior. While it is true that packet sniffers are adept at dissecting the contents of IP and TCP packets, it is common to run across a new protocol that your sniffer doesn't yet support, forcing you to rely on your manual analysis skills to examine the contents of packets.

Let's jump into the material by starting with a discussion about numbering systems and how computers represent data that we can analyze.

Numbering Systems

- Binary: base 2
- Decimal: base 10
- Hexadecimal: base 16
- Every 4 bits equals 1 hexadecimal character



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Numbering Systems

Let's take a closer look at the binary system. Computers are composed almost entirely of billions of tiny little transistors embedded into microchips. Each transistor acts like a little switch, either storing a charge or not and often switching between the two states. Because there are only two states in which any individual transistor can be, it is often convenient to represent numbers in base 2 when dealing with computers. Base 2, also known as binary, is written with only two symbols: a "1" denoting a charge or a positive value and a "0" denoting no charge or the lack of a value.

A single symbol, no matter what value it holds, is referred to as a bit, the fundamental building block of information within a computer. By themselves, however, bits are not large enough to hold information anyone normally would consider interesting. Thus, bits are grouped into collections of 8, referred to as a byte (or octet), the level in which you start to see useful data. A single byte can hold a number from 0 to 255, which the computer could interpret in various ways, such as an integer or as a single character like the letter x or a dollar sign \$.

Human beings have 10 fingers and 10 toes, so it is only natural that our preferred system of counting also has 10 digits. We refer to this numbering system as base 10 because it is based on 10 distinct digits. We are so used to base 10 that many people do not even know that there are other possibilities.

In reality, there are as many different bases as there are numbers with which to express them. The most common numbering systems include decimal (base 10), binary (base 2), octal (base 8), and hexadecimal (base 16).

The decimal number 255 is written in binary as 11111111. A binary 11010101 would be 213 in base 10 (decimal). Notice how the base 10 version is shorter and more compact. This is an example of a general rule: The higher your base, the fewer symbols it takes to write large numbers. Base 2, being the smallest whole number base possible, usually takes a lot of digits to express even fairly small values. This is one of the biggest problems humans have in dealing with binary notation—the digits tend to run together when trying to read them.

Consider the decimal number 53,818. In binary, this would be the rather intimidating 1101001000111010. To make this representation a little easier to read, convention dictates that this long number be broken along 4 bit boundaries when written, like so: 1101 0010 0011 1010. Remember, a byte is equal to 8 bits, so the first two groups of 4 would make up the first byte (1101 0010), and the last two groups would be the second (0011 1010). Still, all these ones and zeros start to look alike after a while, especially when the represented number is large, creating a visual deterrence from distinguishing the placement of the ones and zeros. That is why you often see base 16, or hexadecimal, numbering used instead.

Hexadecimal Numbers

Hex numbers use the digits 0 through 9 just like decimal numbers do, but they need to be able to express 16 possible values for each digit, and the 10 numerals we are used to dealing with every day just are not enough. Hex digits also encompass the Roman alphabet letters A - F, where A = 10 through F = 15. These letters usually are written as capitals, but this isn't a requirement. In hex, each digit can stand for anything from decimal 0 to decimal 15. Two hex digits are the equivalent of eight digits of binary. In other words, it takes only two hex digits to write one byte or 8 bits. In our previous example, the long, intimidating binary number 1101 0010 0011 1010 can be written simply as D23A: byte one as 1101 0010 or as D2, byte two as 0011 1010 or as 3A. Note in this example that the binary value 1101 represents the hex value D, and 0010 represents the hex value 2. This will become important in another few slides when we start to look at fields that are only 4 bits in length.

Whether a number is represented as decimal or binary is usually pretty obvious when seen. Rarely will you mistake a decimal 10 for a binary 10 (usually because binary numbers generally are written in groups of 4 or 8). You might, however, have more trouble distinguishing a decimal 14 from a hex 14 (which, after all is a radically different number!). In text, it is common to do what we have been doing, simply mentioning the number's base when mentioning the number itself. There is another convention you should be aware of, however. Hex numbers are often notated by preceding them with "0x." For example, "0x14" unambiguously is 14 in hex. The "0x" is not really part of the number; it is just a shorthand way of indicating that the value is base 16. In fact, many people find this the most convenient notation, and it is in wide use.

Hexadecimal Representation

2³ 2² 2¹ 2⁰

0	0	0	0	=	0
0	0	0	1	=	1
0	0	1	0	=	2
0	0	1	1	=	3
0	1	0	0	=	4
0	1	0	1	=	5
0	1	1	0	=	6
0	1	1	1	=	7

2³ 2² 2¹ 2⁰ (Hex)

1	0	0	0	=	8
1	0	0	1	=	9
1	0	1	0	=	10 (a)
1	0	1	1	=	11 (b)
1	1	0	0	=	12 (c)
1	1	0	1	=	13 (d)
1	1	1	0	=	14 (e)
1	1	1	1	=	15 (f)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Hexadecimal Representation

This slide gives you a reference point to use when converting between hexadecimal and binary numbers. Remember that 8 bits represent a single hexadecimal value, and 4 bits represent a single hexadecimal character. For example:

0xFAB8 = 1111 1010 1011 1000

Five Tips for Decoding Packets

- Offsets from the beginning of the packet start at 0
- Four bits = 1 hex character
- One byte = 2 hex characters
- Fields can be any length:
 - From one bit to many bytes
 - Fixed length, or variable length
- Fields in one protocol identify the length and contents of others

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

5 Tips for Decoding Packets

With a little practice, manually decoding packets will become second-nature. Before we start decoding packets, there are a few tips that you should be aware of to make this process easier to understand.

Field Offsets

When analyzing the hex contents of a packet, we are often interested in the contents of a specific field that is somewhere past the beginning of the packet. In order to locate the position of the contents of a field we are interested in, we count the number of bytes from the beginning of the file to the desired location of data. When we count the number of bytes or bits, we refer to it as the offset distance from the beginning of the packet. Whenever you see the word "offset", remember that the first byte of the packet is referred to as "offset 0". We always start counting from 0 when locating the contents of the field we are interested in.

Hexadecimal Values

We spent a few slides talking about the use of different numbering systems, converting numbers between different systems, and how those numbers can be represented in many different ways. Remember that binary notation is base 2, decimal notation is base 10, and hexadecimal notation is base 16.

When we evaluate hexadecimal values, each hexadecimal character corresponds to 4 bits (e.g. hex E9 = 1110 1001, where 1110 represents "E", and 1001 represents "9").

Field Length

As we will see when we dive into reading IP and TCP headers, fields in a header can be any length ranging from 1 bit to several bytes. While it is convenient for us to represent values in bytes, it is an inefficient use of space if we can identify everything that is needed in a few bits. Consider the case where we need to indicate if a specific option is enabled or disabled in a packet—using an entire byte to do this would be a significant waste because we only need one bit for the binary operation (on or off). In some cases, protocol designers will opt to use 3 bits of storage for one field, and the remaining 5 for something else. This makes it difficult for us to examine the meaning of the fields in hexadecimal notation, so we have to convert to binary format to evaluate the bits-wise fields independently.

The majority of the fields we will be examining in this chapter are fixed-length - they always use the same number of bits to store the data for a given field. For instance, the Time to Live field in the IP header will always be 1 byte in length, the destination port in the TCP header will always be 2 bytes in length, and the TCP header length will always be 4 bits in length. Some fields will be variable length, using a field inside of the variable-length field to determine the length of the data. A good example of a variable length field is the TCP Options field. Some operating systems will include parameters to indicate what TCP options are in use, using a flag to indicate the specific TCP option, another field to indicate the length of the specific option, and finally the option(s) that follow. This provides protocol designers a way to include functionality to identify what TCP options they use, without requiring a large fixed-length field to identify all possible options, when most of them aren't in use.

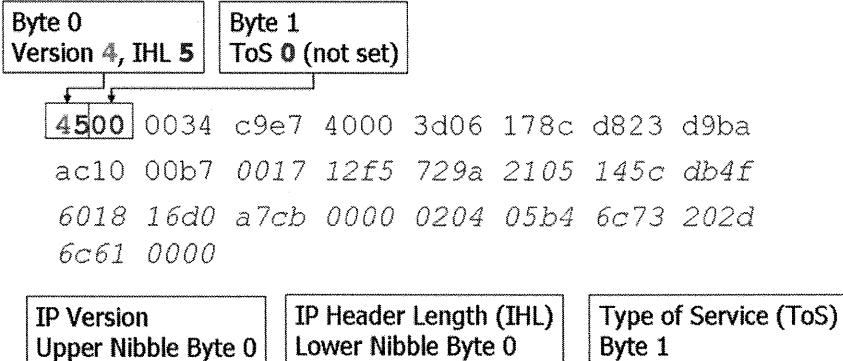
Relationships Between Headers

When packets are generated, one layer's header becomes another layer's data. For example, an Ethernet packet has a data payload that contains an IP header. The IP header that is the data payload for the Ethernet packet also has a data payload that contains a TCP header. The TCP header that is the data payload for the IP header has a data payload that contains a HTTP header, and so on.

Each parent header contains information about the child header including the length of the child packet and sometimes the type of embedded protocol. For instance, in an ICMP echo request or "ping" packet, the IP packet will have a value of 1 in the IP Protocol field, indicating it has an embedded ICMP packet header. The IP packet header will also indicate the length of the IP header, and the total length of the packet. Subtracting the two values will tell us the embedded protocol packet length.

For the exercises in this chapter, remember that we will use the information collected in each portion of the packet to decode the remainder of the packet. Even if we are interested only in the embedded protocol (for example, HTTP), we have to examine the parent headers to identify where the embedded protocol portion of the packet starts.

Decoding an IP Header (1)



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Decoding an IP Header (1)

This slide introduces a sample packet, displayed in hexadecimal format. As humans can read packets and other data in hexadecimal format by dissecting the different portions of the packet, converting the values to decimal or binary as needed, and evaluating the contents. This packet was captured with the packet sniffer tcpdump and is showing us the contents of an IP packet. Note that we aren't examining the contents of the Ethernet packet header that would precede the IP header information for this exercise, although you should be aware that other packet captures may present that information to you.

The same slide format will be used for the next several slides - note the bottom of the slide identifies the fields we are examining, indicating the contents of the field ("IP Header Length"), and the byte offset from the beginning of the packet header. Above the packet contents we have identified the field contents with the hex value ("0x45"), and the corresponding values for the fields ("Version 4, IHL 5").

Each grouping of hexadecimal values corresponds to 2 bytes of data (e.g. "178c" represents 2 bytes of the packet; "d823" represents the next 2 bytes of data). Remember that a hexadecimal value of 8 bits (1 byte) is represented with two characters ("d8"), and a single hex character is represented with 4 bits ("d").

This slide starts our analysis of the packet by identifying the IP Version, IP Header Length, and Type of Service fields in the IP packet header.

The IP Version field, as the name implies, indicates the version of the IP protocol being used for this packet. Notice that it is the first part of the packet that would be processed by the receiving station - this is so that the receiving station knows what header format to use when processing the packet. In this case, the IP version is Version 4, stored in a field 4 bits in length. For IPv6 packets, the contents of this field would be "6". Once the receiving station identifies that this is an IPv4 packet, they can make assumptions about the rest of the locations of fields in the packet header based on how IPv4 packets are defined in the RFC's. This field has an offset of 0, meaning that it is located at the beginning of the packet header, and extends for 4 bits. This is convenient for us, because 4 bits corresponds to a single hex character; the byte represented in hexadecimal as "45" indicates an IP version of "4".

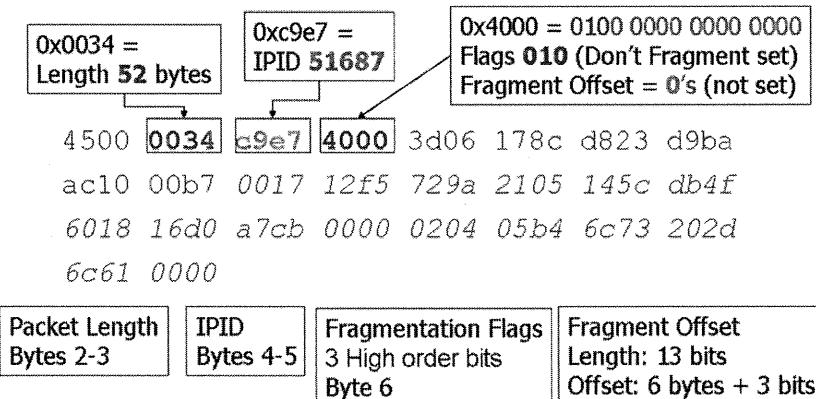
The next field is the IP Header Length field, abbreviated as "IHL". This field is also 4 bits in length, and has an offset of 4 bits. Counting from 0, this is the 4th bit offset from the beginning of the packet header. The IHL field tells us how long the IP header is in 32-bit words. To calculate the length of the IP header, multiply the IHL value by 4 (4 bytes = one 32-bit word). This field is contained in the low order nibble of byte 0, immediately following the IP Version field. Like the IP Version field, the IHL field corresponds to a single hex character; the byte represented in hexadecimal "45" indicates an IHL of "5".

With the IHL field, we can calculate the length of the IP header. Because the IHL field is the length of the IP header in 32-bit words, we multiple the IHL value by 4 (4 bytes in a 32-bit word). In this example, the IP header length is 20 bytes ($5 * 4$). Notice in this slide that the first 20 bytes of the packet are darker than the trailing bytes. The darker values represent the IP header, with the trailing bytes representing the Layer 3 protocol. We'll continue examining the fields in order to examine the embedded protocol after we finish examining the IP header.

Following the IHL field, we have the IP Type of Service field (ToS). This field is one byte in length and is represented by byte 1 (the second byte of the trace, because we start counting from 0). The ToS field is used to convey the priority for delivering this packet in reference to other IP packets with varying ToS values. For example, real-time applications like voice over IP and video over IP would have a higher value in the ToS field to give it preference over other protocols such as HTTP or FTP. In this case, there is no precedence set for the delivery of the packet, so the ToS field is set to 0x00 (zero).

In this slide we discussed the techniques we can use to read a packet header, and examined the contents of the IP version, IP Header Length, and Type of Service fields. Let's continue our analysis of this packet by looking at the remaining fields in the IP header, as well as the embedded protocol header.

Decoding an IP Header (2)



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Decoding an IP Header (2)

This slide examines the Total Packet Length, IP Identification, and Fragmentation in the IP packet header.

Total Packet Length is the next field in the IP packet header, which indicates the total length of the IP packet, including the payload or data portion of the packet. This field has a length of 2 bytes, and is contained in bytes 2 and 3. Unlike the IHL field, the Total Packet Length field is represented in bytes, not 32-bit words. When determining the total packet length, we simply count two bytes offset from the beginning of the packet, and convert the next two bytes to decimal format to identify the total length of the packet. In this case, the Total Packet Length field has a value of 0x0034, which is the hexadecimal equivalent of 52, indicating this packet is 52 bytes in length.

Before we continue to look at the next field in the packet header, let's take a look at the order of the fields we have covered so far. The first 4 fields in the IP packet header are (in order) IP Version, IP Header Length, Type of Service, and Total Packet Length.

The order of these fields may seem arbitrary at first, but when we take a closer look, we can see that the IP protocol designers put a lot of careful thought into the structure and order of the fields. Imagine what a router must do when it receives a packet from a network interface.

The router knows it is receiving an IP packet because the Layer 2 protocol (Ethernet, HDLC, 802.11/802.2, etc.) indicated it was delivering an IP packet in the embedded protocol field. The router doesn't yet know what version of IP protocol it is receiving, so the first thing it has to do when it receives the packet is examine the IP version field. Because this is the first field in the IP header, the router can do this very quickly.

The next thing the router has to do is determine the length of the IP header so it can allocate enough memory to hold the packet header. Immediately following the IP Version field is the IP Header Length field, which tells the router the size of the packet header.

After allocating enough memory for the IP packet header, the router needs to know how important this packet is. Should the router continue processing this packet, or should it spend time processing other packets first? To answer this question, the router checks the Type of Service field to identify the priority for processing this packet and schedules how it processes packets accordingly.

When the router is ready to process the packet, it needs to allocate enough memory to store the entire packet in memory to process and deliver it to the destination. To do this, the router needs to know the total length of the packet. This field follows the ToS field, and tells the router the exact length of the packet so it can allocate the appropriate amount of memory to store the packet.

When we identify and dissect the remaining fields in the IP packet header, keep thinking about the packet from the perspective of a router or a host that is processing the packet. You will consistently find that the IP packet header is structured and sized optimally for performance and ease of processing.

The field that follows the Total Packet Length field indicates the IP Identification (IPID) value for this packet. Remember that the IPID field is used to associate multiple fragments of packets together that make up a single IP packet when reassembled. This field can also be used to invisibly port-scan a server too, but we'll get to that later! The IPID field is 2 bytes in length and is contained in bytes 4 and 5. In this example, the IPID field has a value of "0xc9e7", which can be represented in decimal format as 51687.

Before we start examining the contents of the last fields in this slide, let's review the concept of fragmentation. You can think of a fragment as a portion of something -- by itself it doesn't mean much until you have all the pieces put together. Ever drive down the highway and see a portion of a modular home on a tractor-trailer truck? By itself, the garage or the living room or the bathroom aren't very useful (typically with two sides and no roof!), but when assembled together, they make a house. Fragmentation of IP packets works in a similar fashion. If a transmitting station wants to send a single IP packet that is too big for the destination network, the host will chop the packet up into several pieces and transmit each one individually.

Each portion of the original packet (including the first and last pieces) is a fragment. When the receiving station finishes collecting all the fragments, they are reassembled to become a single packet that can be processed. To accomplish IP fragmentation, the IP header has the header flags and fragment offset fields to use in reassembling the packet.

The IP Header Flags field is used to indicate the presence of packet fragmentation and whether a packet should or should not be fragmented. This field has an offset of 6 bytes and a length of 3 bits. This is the first field in the IP header that does not end on an even 4-bit boundary. Because of this, we have to convert the hexadecimal value into a binary value to see what bits are set. To do this, we count to the 6th byte offset from the beginning of the IP header and examine the contents of the field. In this example, we see the entire byte is 0x40. We can convert the field 0x40 to binary format:

$$0x40 = 0100\ 0000$$

Because we are interested in the first three bits of this value, we can use the first three bits ("010") as the binary contents of the IP Header Flags field. Matching these binary values to the order of the bit-fields in the IP Flags field, we can determine what flags are set. In this case, "Bit One" represents the most-significant or leftmost bit:

Bit One	Reserved	0
Bit Two	Don't Fragment	1
Bit Three	More Fragments	0

In this case, the binary value for the IP Header Flags field ("010") indicates that the reserved field is not set, the don't fragment field is set, and the more fragments field is not set.

When assessing the contents of bitwise fields, convert the hexadecimal value for the field that includes the fields you are interested in to binary format. Then, use only the bits that map to the field you are assessing.

The next field in the IP Header is the Fragment Offset field. The Fragment Offset field is used for fragmented packets to tell the receiving station how to reassemble all the fragments once they are received. The fragment offset value indicates the number of bytes from the beginning of the total packet that this fragment should be placed. This field is 13 bits in length, and has an offset value of 6 bytes and 3 bits (immediately follows the IP Header Flags field).

Reading the contents of this field is just like reading the contents of the IP Header Flags field. We simply count 6 bytes offset and read the 2 bytes of data in hexadecimal format. In this case, the value is 0x4000. Then we convert the hexadecimal value 0x4000 to binary, and evaluate only the bits we are interested in:

$$0x4000 = 0100\ 0000\ 0000\ 0000$$

Because we are looking at 13 bits for this field, we discard the first 3 bits (used for IP Header Flags) and evaluate the remaining bits. In this case, all the bits are 0, which indicates that this field is not set.

Decoding an IP Header (3)

0x3d =
TTL 61

0x06 =
Embedded Protocol 6 (TCP)

4500 0034 c9e7 4000 **3d****06** 178c d823 d9ba
ac10 00b7 0017 12f5 729a 2105 145c db4f
6018 16d0 a7cb 0000 0204 05b4 6c73 202d
6c61 0000

Time to Live (TTL)
Byte 8

Embedded Protocol
Byte 9

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

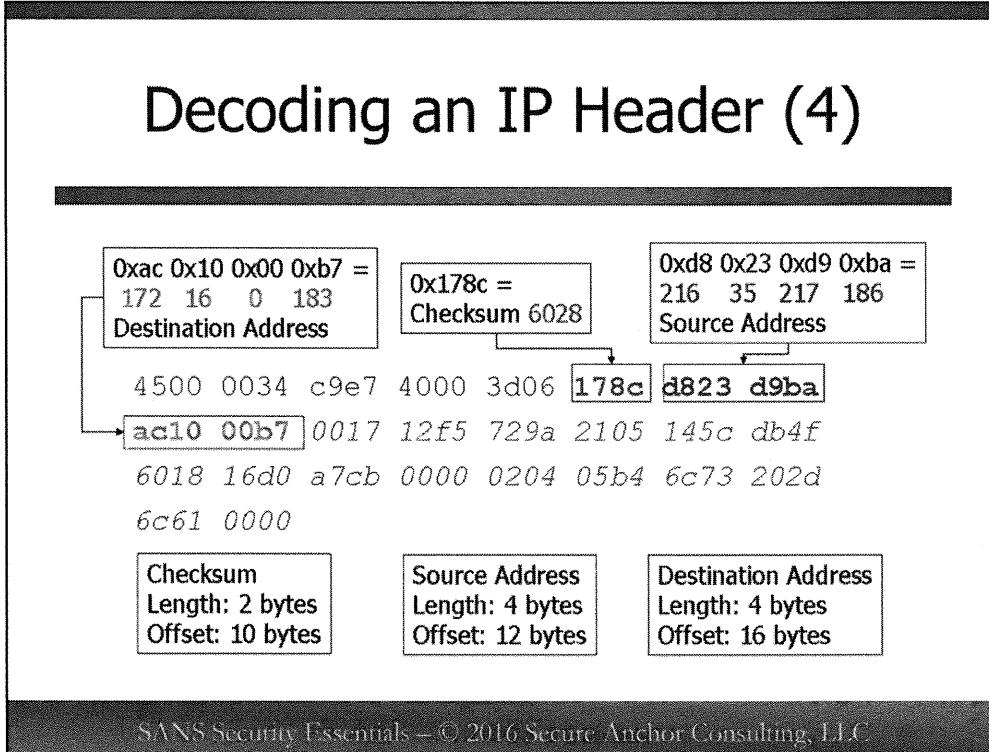
Decoding an IP Header (3)

This slide examines the Time to Live and the Embedded Protocol fields in the IP packet header.

The next field we are going to examine is the Time to Live (TTL) field. Remember that the Time to Live field is used to detect routing loops and is decremented by one for each hop on the way to its destination. Counting the 8th byte offset, we see the TTL field has a value of "0x3d", which can be represented in decimal format as 61.

Next, the Embedded Protocol field indicates the protocol type for the IP packet payload; we use this field to determine if this packet is using TCP, ICMP, UDP, GRE, or any number of other protocols. The Embedded Protocol field is one byte in length and is located at 9 bytes offset. In this example, the 9th byte offset is "0x06". Using the quick-reference on your TCP/IP and tcpdump Reference Guide, we see that this indicates the embedded protocol is TCP. Note that the IPv6 header also uses an embedded protocol field, but this field has been renamed to "Next Header".

Decoding an IP Header (4)



Decoding an IP Header (4)

This slide examines the IP Header Checksum, the Source Address and the Destination Address fields in the IP packet header.

In order to prevent a device from trying to process a corrupted packet, the IP header includes a Checksum field that is used to store a 16-bit one's complement sum of the IP header data (the checksum does not cover payload data). When the IP packet is first assembled and each time it is changed en-route to the destination, the checksum value is re-calculated to ensure the packet hasn't been accidentally corrupted. Note that this feature is not intended as a security mechanism, because the Checksum field can be manipulated with bit-flipping attacks. Instead, it is used to identify accidental packet corruption. This field is 2 bytes in length with an offset of 10 bytes. In this example, the IP header checksum is 0x178c, which can be represented in decimal as 6028.

The next two fields in the IP header are the source and destination addresses. These fields are both 4 bytes in length, with the source address appearing first with an offset of 12 bytes, and the destination address appearing next with an offset of 16 bytes.

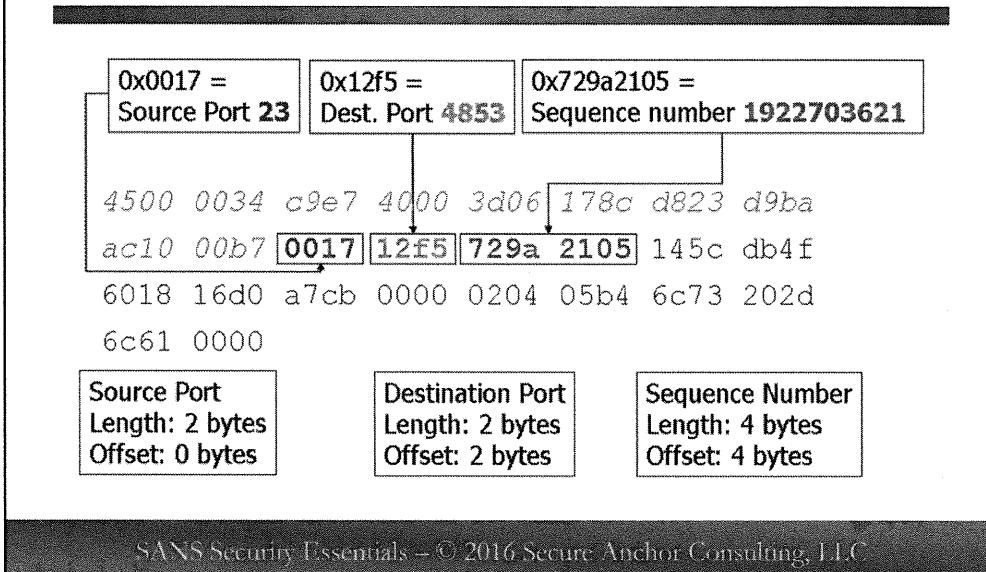
When we use IP addresses, we typically see them represented in dotted-decimal format consisting of four decimal values separated by periods. Each of the four values in an IP address can be in the range of 0 - 255. This is a convenient representation for humans, but we can also represent IP addresses as four hexadecimal values, with each hexadecimal value representing one of the four decimal values in dotted-decimal notation.

For example, the IP address "65.173.218.106" can be represented as "0x41ADDA6A"(0x41 = 65, 0xAD = 173, 0xDA = 218, 0x6A = 106). Similarly, an IP address represented in hexadecimal format as "0xE0000005" can be converted to dotted-decimal format as 224.0.0.5 (0xE0 = 224, 0x00 = 0, 0x00 = 0, 0x05 = 5).

In this case, we see the source address at offset 12 is "0xd823d9ba", which is represented as 216.35.217.186 in dotted-decimal format. The destination address located at offset 16 is "Oxac1000b7" which is represented as 172.16.0.183.

So far, we've examined the IP Version, IP Header Length, Type of Service, Total Length, IP Identification, IP Flags, Fragment Offset, Time To Live, Embedded Protocol, Checksum, Source Address, and Destination Address fields. By this time, you're probably a pro already at decoding packets! Just for some added reinforcement of the material, let's continue our analysis to examine the fields in the embedded protocol for this packet, a TCP packet.

Decoding a TCP Header (1)



Decoding a TCP Header (1)

In our analysis of the IP packet header, we saw the embedded protocol indicate the payload is a TCP packet. We can use the TCP header information in the TCP/IP and tcpdump Pocket Reference Guide to examine the TCP packet header fields to find out more about this packet. Let's start our analysis by examining the source port, destination port, and sequence number information for this packet.

Notice how the example packet in this slide has the trailing 30 bytes of the packet darker than the fields we analyzed as the IP header. The darker hexadecimal values represent the IP payload (which we know is the TCP packet). Now that we are assessing the contents of the TCP packet, our offset values return to 0 to indicate a relative offset from the beginning of the TCP header. We could use an absolute offset from the beginning of the IP packet for the location of TCP fields, but this value would change if the size of the IP header changed (which happens when the IP header uses IP options, for example). For the remainder of the slides assessing the TCP header, remember that the offset value starts from the beginning of the TCP header, which is indicated in the slide with darker text.

The first field in the TCP header is the Source Port. This field indicates the source port for the connection, but it is not necessarily the port used by the client initiating the connection. If the server is responding to a connection, this value will be the server port number. The TCP Source Port field has a length of 2 bytes, and an offset of 0. In this example, the source port is "0x0017", which is represented in decimal format as 23. This port is commonly associated with the Telnet protocol.

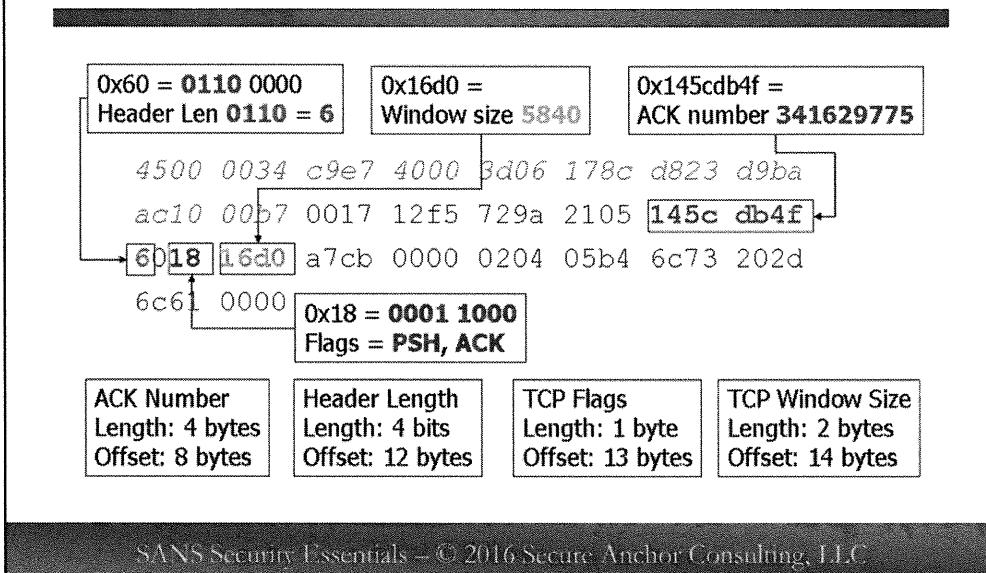
The second field in the TCP header is the Destination Port. This field indicates the destination port for the connection; like the source port, this can be the client port or the server port number depending on whether the traffic flow is from the client or the server. The TCP Destination Port has a length of 2 bytes and an offset of 2 bytes. In this example, the destination port is "0x12f5", which is represented in decimal format as 4853.

Following the TCP Destination Port is the TCP Sequence Number field. Sequence numbers are used in a TCP session to keep track of "flow" between a client and a server, establishing a mechanism to identify multiple

segments of data that are transmitted in a single TCP stream. Sequence numbers are used to establish an order for the transmission of multiple segments of data to a recipient. This field is 4 bytes in length, with an offset of 4 bytes. In this example, the sequence number is "0x729A2105", which is represented in decimal format as 1922703621.

So far, we've identified that this packet has a source port of 23, a destination port of 4853 and a sequence number of 1922703621. Because the source port is 23, we might guess that this packet is from the server to a client, but this isn't always the case. Let's examine the other fields to determine what this packet is for.

Decoding a TCP Header (2)



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Decoding a TCP Header (2)

This slide examines the TCP Acknowledgement Number, TCP Header Length, TCP Flags, and TCP Window Size fields in the TCP packet header.

The TCP Acknowledgement Number is used to ensure the reliable delivery of TCP packets. This field is used to acknowledge the receipt of data sent in the form of the last received sequence number + 1. This way, the host that sent data knows the recipient has properly received data through the acknowledgement of the previous sequence number. This field is 4 bytes in length, and has an offset of 8 bytes. In this example, the acknowledgement number is "0x145cdb4f", which can be represented in decimal format as 341629775.

The next field is the TCP Header Length field. Like the IP Header Length field, the TCP Header Length field represents the size of the TCP header in 32-bit words (4 bytes). Remember that the TCP and IP header length fields only indicate the length of the header - they do not indicate the length of any data included with the packet. This field is 4 bits in length, with an offset of 12 bytes. Because this field is not an even number of bytes in length, we have to convert the value to binary and examine the 4 most-significant bits (leftmost bits), or examine the most-significant hexadecimal character. In this case, the byte-long value is "0x60", which is "0110 0000" in binary. Because we are only interested in the 4 most-significant bits, we convert "0110" to decimal, which is 6. Because the TCP Header Length is represented in 32-bit words, we multiply this value by 4, which indicates the TCP header length is 24 bytes.

The four bits that follow the TCP Header Length field are reserved for future use. These fields should always be set to "0" per RFC's, but they may be unintentionally set by some operating systems that do not properly initialize allocated memory.

The next field we will examine is the TCP Flags field. This field is used to indicate the purpose of a TCP packet; by setting a bit-wise field, the TCP packet indicates a new connection, an acknowledgement of data received, a connection tear-down, urgency of data processing, and several other settings. For now, we'll just examine the contents of the field in order to understand how to decode the TCP packet header. This field is 1 byte in length and has an offset of 13 bytes.

In order to evaluate the bits that are set in the TCP Flags field, we must convert the hexadecimal value to binary. In this example, the TCP Flags field has a value of "0x18". When we convert the value to binary, we get "0001 1000". The TCP Flags field is formatted as follows:

Bit One*	Congestion Window Reduced (CWR)
Bit Two	Explicit Congestion Notification Echo (ECN-Echo)
Bit Three	Urgent Pointer is Valid (URG)
Bit Four	Acknowledgement Set (ACK)
Bit Five	Pass Data to Application ASAP (PSH)
Bit Six	Reset the Connection (RST)
Bit Seven	Synchronize Set (SYN)
Bit Eight	Sender if Finished (FIN)

* Bit One represents the most significant, or leftmost bit.

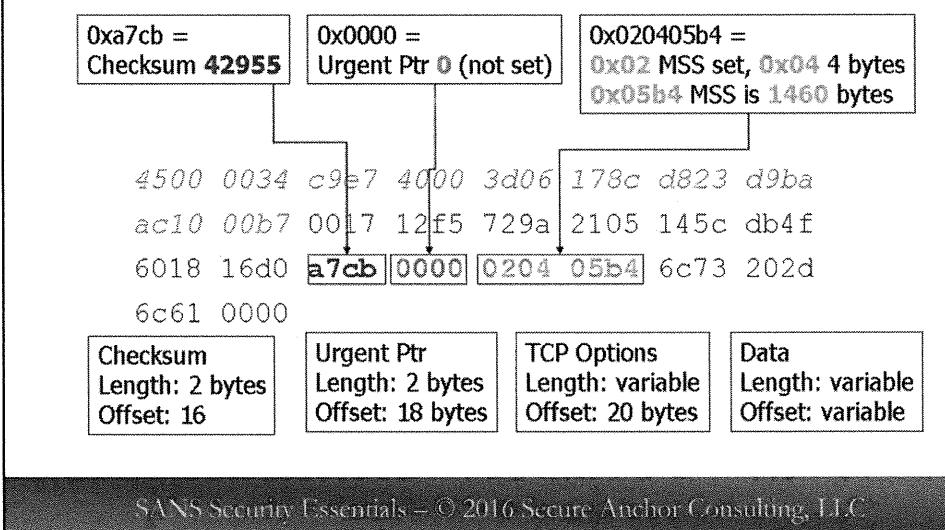
Mapping these flags to the fields in this example, we get:

| C | E | U | A | P | R | S | F |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | = PSH set, ACK set

Remember, when assessing the TCP Flags field, convert the contents of the 13th byte offset to binary, and map the flags accordingly.

The TCP Window Size field is used to indicate the number of bytes that can be transferred to the destination before they must be positively acknowledged with an ACK. The window size will increase on reliable networks that can accommodate large quantities of traffic without any packet loss. On networks that are unreliable with frequent packet loss, the window size will be smaller to deliver fewer packets before requiring a positive acknowledgement. The TCP Window Size field is 2 bytes in length, with an offset location of 14 bytes. In this example, the TCP Window Size is "0x16d0", which can be represented in decimal format as 5840. This indicates the transmitting station is willing to accept 5840 bytes of data before sending an ACK message.

Decoding a TCP Header (3)



Decoding a TCP Header (3)

This slide examines the TCP Checksum, TCP Urgent Pointer, TCP options, and payload data in the TCP packet header.

Like the IP header, the TCP header has a checksum field to prevent accidental corruption of the header fields. This field is 2 bytes in length, and is located at 16 bytes offset. In this example, the TCP Checksum field is "0xa7cb", which is represented in decimal format as 42955.

The Urgent Pointer field is used to indicate the location of "urgent data" to the receiving station. This feature is commonly used with terminal emulation protocols including telnet to indicate the presence of "CTRL/C" indicating a break that should be processed urgently. The Urgent Pointer field points to the last byte of urgent data that should be processed by the receiver. This field is 2 bytes in length, with an offset value of 18 bytes. In this case, the Urgent Pointer field is all 0s, which indicates it is not set.

The last field in the TCP header is used to indicate TCP options. This field is variable in length, ranging from 0 to 40 bytes in length. Let's examine how we can calculate the length of this and other variable-length fields using the information that is provided to us in the packet. To analyze the contents of this field, we have to determine the length of the TCP options to ensure we don't confuse TCP options with payload data. The TCP header doesn't include a field to indicate the length of TCP options; instead we must calculate the value by using the TCP Header Length field and the known length of a TCP header without options.

With no TCP options set, the TCP header length is 20 bytes, or 5 32-bit words (20/4). We can calculate the length of the TCP Options field by subtracting the value of the TCP header length by 5. In this case, we saw the TCP header length was 6 32-bit words. Subtracting the two, we can determine the TCP options are 1 32-bit word in length, or 4 bytes.

The TCP Options field is always at an offset value of 20 bytes. Knowing that the TCP Options field length is 4 bytes, we can see the TCP options in this example are "{0x020405b4}". We can evaluate this field further, where the most-significant byte (leftmost) "0x02" represents the specific option that is set (Maximum Segment Size), "0x04" represents the number of bytes total for this option (including the TCP option tagged parameter, 4 bytes), and the MSS value is set to "0x05b4", or 1460 bytes. This is standard for an Ethernet network with a Maximum Transmission Unit (MTU) size of 1500 bytes.

Calculating Variable Length Fields

- TCP options and the length of payload are variable width fields
- We calculate some field lengths using other header information and fixed lengths

TCP options length = (TCP header length - Min. TCP header length)

Length of packet payload = IP total length - (IP header length + TCP header length)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Calculating Variable Length Fields

To analyze the contents of variable-length fields, we need to examine the contents of other fields in the packet and use fixed-length information, such as the minimum size of a header.

Variable length fields are common in many protocols, allowing the transmitter to use the protocol to transmit additional information when it is available, without having to waste space with a fixed-length field that may not be used frequently. When reading and decoding the contents of packets, we need to carefully calculate the length of fixed-length fields to continue assessing the contents of the packet. Let's look at an example taken from the packet example used in this module to assess the length of the TCP Options field.

To analyze the contents of the TCP Options field, we have to determine the length of this variable-length field to ensure we don't confuse the TCP options with the payload data that follows TCP options. The TCP header doesn't include a field to indicate the length of TCP options. Instead, we must calculate the value by using the TCP Header Length field and the known length of a TCP header without options. With no TCP options set, the TCP header length is 20 bytes, or five 32-bit words (20/4). We can calculate the length of the TCP Options field with the following formula:

$\text{TCP options length} = (\text{TCP header length} - \text{Min. TCP header length})$

In this formula, the TCP option's length is equal to the size of the reported TCP header length in the TCP Header Length field minus the size of a TCP header with no options (20 bytes). Remember that the TCP Header Length field is represented in word-size, so we need to multiply the field contents by 4 to represent the length in a number of bytes.

Next, we calculate the length of the packet payload. We calculate this value using the following formula:

$\text{Length of packet payload} = \text{IP total length} - (\text{IP header length} + \text{TCP header length})$

In this formula, the length of the packet payload is equal to the contents of the IP Total Length field minus the sum of the IP Header Length field and the TCP Header Length field. This would tell us the length of payload data that is carried in the TCP packet, which we would use as a starting point to continue our analysis to identify the contents of the embedded protocol.

Now that we know how to assess the length of variable-length fields, we can return to our packet analysis and examine the contents of the TCP Options field and TCP Payload field.

Decoding a TCP Header (4)

$$(\text{TCP Header Length} - \text{Min. TCP Header Length}) = \text{TCP Options Length}$$
$$(6 * 4) - 20 = 4$$

$$\text{IP Total Length} - (\text{IHL} + \text{TCP Header Length}) = \text{Payload Length}$$
$$52 - ((5 * 4) + (6 * 4)) = 8$$

4500 0034 c9e7 4000 3d06 178c d823 d9ba

ac10 00b7 0017 12f5 729a 2105 145c db4f

6018 16d0 a7cb 0000 0204 05b4 6c73 202d

6c61 0000

Payload
Length: 8
Offset: 24 bytes

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Decoding a TCP Header (4)

Here we have the formulas for calculating the variable-length TCP Options field and TCP Payload field in the TCP packet. Our calculations have to follow a very simple processing order to avoid positional errors.

First, we calculate the TCP options length by subtracting the TCP header length value from the fixed minimum TCP header length, converting the TCP header length value from a number of words to a number of bytes by multiplying it by 4. Subtracting the two, we see the TCP options length is 4 bytes.

As a review, the TCP Options field is always at an offset value of 20 bytes. Knowing that the TCP Options field length is 4 bytes, we can see the TCP options in this example are "0x020405b4". We can evaluate this field further, where the most-significant byte (leftmost) "0x02" represents the specific option that is set (Maximum Segment Size), "0x04" represents the number of bytes total for this option (including the TCP option tagged parameter, 4 bytes), and the MSS value is set to "0x05b4", or 1460 bytes. This is standard for an Ethernet network that Maximum Transmission Unit (MTU) size of 1500 bytes.

Next, we calculate the length of the TCP payload. Because this is the last field in the packet, the length of the TCP payload should match the number of remaining bytes left to assess. This can be manipulated by an attacker, however, so we should perform the analysis to identify the reported TCP payload length and compare it to the observed length by counting the number of remaining bytes.

Because the TCP payload follows a variable-length field (the TCP Options field), we need to calculate the length of the previous field and add that to the last known fixed offset. In this case, the last known fixed offset field is the beginning of the TCP Options field at 20 bytes offset. Because we calculated the TCP Options field to be a length of 4 bytes, we add that length to the last known fixed offset to identify the beginning of the TCP payload at an offset of 24 bytes.

We can calculate the payload length by calculating the sum of the IP Header Length and the TCP Header Length, multiplying the value by 4 to produce a number of bytes, and subtracting it from the reported IP Total Length. The example packet on this slide has an IP Header Length of 20 bytes, a TCP Header Length of 24 bytes and a Total Length of 52 bytes:

$$52 - (20 + 24) = 8 \text{ byte payload}$$

The trailing 8 bytes of the packet indicate packet payload. In this case, the hexadecimal values "0x6c73 0x202d 0x6c61 0x0000" can be converted with an ASCII chart to "ls -la". Note that the trailing NULL bytes ("0x00") are included to pad the packet to a word-boundary (evenly divisible by 4).

At the end of all this analysis, what do we know about this packet? It is a TCP packet with 6 bytes of payload from a Telnet server to a client, containing the string "ls -al". This is probably the echo from the telnet server as someone entered the Unix command to list all the files in the directory in long format. This packet is also acknowledging the receipt of data from a previous packet with the ACK flag set, and is advertising a Window size of 5840 bytes. Not bad for a few minutes work.

Reading Packets Summary

- Reading packets is an acquired skill
- Necessary when protocol decoders are not available
- Fields in one header will identify length and content of others
- Save your TCP/IP Reference Guide!
- Practice makes perfect

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Reading Packets Summary

This section has reviewed the techniques used to decode packets. We spent several slides decoding the contents of an IP and a TCP packet, but the skills you've learned are applicable to decoding any type of packet. All you need is a reference to the header layout and contents of the fields and you will be able to decode any type of packet. Remember that reading packets is an acquired skill; it is a long and tedious process to decode a packet manually when you first start. Like many skills, as you improve through practice, it will become easier to read packets, count offsets, and decode the contents of fields.

Summary

- Sniffers are invaluable network analysis tools, but they must be protected
- **tcpdump:** Free sniffer; analyzes IP, ICMP, TCP, and UDP traffic

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

You should be familiar with the legitimate traffic that regularly appears on your network so you can recognize anomalies when they occur. The sniffer is an invaluable tool for the network administrator because it shows exactly what is happening on the network at a given time. Because of the multitude of sniffer products available, there should be one for you regardless of your budget and your purpose.

Remember, however, that attackers can use sniffers already installed on victim hosts, but they will more likely use the sniffers that often come bundled with rootkits. Even so, sniffers should not be installed where they're not needed, and special privileges should be required to use them. Attackers mainly are after passwords and sensitive data, so strong cryptography is a powerful countermeasure against enemy sniffing.

tcpdump is a free sniffer with a powerful filtering language that makes it easy to analyze just about any type of traffic. It can be tuned with various command-line options; for example, -n disables DNS lookups. tcpdump reads in only the first 68 bytes of each packet by default, but the -s option lets you adjust that size.

Though tcpdump has a fairly standard default output format, it varies a little depending on the protocol being displayed. UDP datagrams are often, but not always explicitly labeled. TCP segments are easy to spot because TCP flags, sequence numbers, acknowledgment numbers, and options are always displayed. ICMP datagrams are always labeled as such, so they're always easy to recognize.

Recognizing anomalies and performing in-depth analysis of a packet, though, often requires going beyond the default output format. The -s option in combination with -x, which dumps packets in hexadecimal, can be used to display the entire raw packet. To store a raw dump for later analysis, use -w. Some time later you can use -r with a filter and any other option to read in and analyze the raw dump.

Module 6: Wireless Network Security

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 6: Wireless Network Security

This section intentionally left blank.

Wireless Network Security

SANS Security Essentials IV: Secure Communications

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Introduction - Wireless Network Security

What does wireless networking mean to you? To some it means complete and total mobility, the freedom to make calls, send e-mail, or surf the Internet from anywhere in the world without being chained to a wire. Perhaps you see it as a cost-effective way to extend your corporate network or maybe as a way to increase productivity on the factory floor. Others see wireless as an opportunity to take advantage of weak security measures for anonymous Internet access, or to avoid the restrictions of your corporate firewall when attempting to exploit servers and hosts.

Objectives

- Wireless Overview
- Bluetooth
- Zigbee
- 802.11
- Wireless Security

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

In this module, we cover an often misunderstood, if not overlooked aspect of deploying and utilizing wireless networks: security. As an individual tasked with securing your organization's resources, we present the key information elements that will allow you to understand the risks and limitations of wireless networking. We discuss how wireless networks are being used in enterprises today and some of the common architectures and protocol implementations of wireless networking. We also review common misconceptions about wireless security, the top five risks of wireless networks, and look at some recommendations for planning a secure wireless LAN.

Wireless Overview

The student will have a basic understanding of wireless technologies.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Wireless Overview

This section intentionally left blank.

Popular Wireless Devices

- Mobile phones
- Laptops
- Tablets
- HVAC control units
- Medical devices

Wireless Market in Growth Cycle for Enterprises, Consumers

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Popular Wireless Devices

In recent years, the adoption of wireless technology has grown significantly. With the cost of wireless adapters and access points consistently falling, enterprise organizations and consumers alike have readily accepted wireless technology for all of their mobile devices including PDAs, mobile phones, handheld computers, laptops, and more.

Interestingly, wireless technology is spreading from the traditional computing environment (laptops, desktops, IP networks) into different vertical markets with technologies such as Bluetooth and ZigBee. It's not uncommon for previously unlikely devices such as HVAC controllers to utilize wireless technology for management functions.

If you look at this surge in popularity from a marketing product life cycle (introduction, growth, maturity, and decline) perspective, wireless has transitioned from the introductory phase and is firmly entrenched in the growth cycle. The key indicators that we are seeing a growth cycle for wireless networks are the emergence of competition, aggressive marketing, and wider availability of products.

Wireless Advantages

- Wiring takes time and money; wireless drastically reduces these costs
- Users can access the network from anywhere
- Mobility and connectivity
- Usable in environments where wiring is difficult:
 - Historic buildings
 - Factories, assembly lines, warehouse floors, hospitals, and financial trading floors
 - Temporary networks such as exhibitions

SANS Security Essentials - © 2016 Secure Anchor Consulting, LLC

Wireless Advantages

Why wireless? Perhaps the better question would be, why not wireless? As we have shown, freedom is the siren song of wireless networks. By itself, the ability to be completely mobile while staying connected is a great reason to deploy wireless technologies, but there are other compelling examples that also warrant attention.

Enabling connectivity where it simply was not possible before is a very attractive reason to deploy wireless networks. Factories and warehouses are a prime example of wireless technologies enabling extended connectivity. Before the viability of mobile computing, it would have been a labor intensive, time consuming, and very expensive endeavor to wire a factory floor for network connectivity. Another example would be to extend your network over obstacles that make wire connections difficult, such as a temporary need to electronically traverse an airport runway. For most organizations, wiring for connectivity within budget simply might not be feasible. Using wireless technology, it is now simply a matter of installing a few pieces of equipment. Networks can be extended, quite literally, in a matter of minutes.

Increased productivity and ease of use are obvious choices but should not be underestimated. Users can be connected and mobile, meaning they can roam the building with their laptop with little or no disruption in service. For example, Josephine User is scheduled to give a presentation in the west conference room. Normally, she would have to log off the network, shut down her machine and disconnect the network cable.

It would be nice to simply pick up the laptop and move to the conference room without having to log off and shut down. It also increases productivity in the process!

Some companies are offering wireless access as a value-added service for their customers. Want to check your e-mail while sipping your mocha latte at the local coffee shop? Need to dump some stock before getting on that plane? Not a problem! Airports, coffee shops, shopping malls, and other areas where large groups of people gather are quickly realizing they can add value or service to their customers at relatively little cost by installing wireless access points to the Internet.

Finally, temporary networks, such as those used at exhibitions and conferences, are ideally suited for wireless technologies. Because wireless networks require relatively few components, they can be set up and torn down quickly. It is becoming increasingly rare to attend a technical trade show or conference without being afforded some type of wireless network access.

Vertical Markets

- Healthcare
- Financial
- Academia
- Factories/Industrial
- Retail
- Wireless Internet Service Providers
- Mobile hot spots

Many markets benefit from wireless, making it popular with users.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Vertical Markets

As the wireless market continues to grow and evolve, we are seeing targeted solutions for a multitude of industries. Some of the biggest vertical markets include healthcare, financial, academia, industrial, and retail.

Healthcare

Hospitals are constantly faced with the daunting task of lowering operating costs while still giving their patients superior quality and personalized care. For many hospitals, fees for services are pre-determined and are often less than the actual monetary amount needed to recoup the cost of performing the service. It does not take a brain surgeon to figure out that operating at a financial loss does not lend itself well to longevity in the marketplace. How can hospitals lower their costs, attract more patients, and get those patients through the system quickly while still maintaining a personal relationship? The answer to that question, as evidenced by increased deployment in the healthcare industry, is wireless networks.

Hospitals are quickly realizing there is a huge ROI (Return on Investment) in being able to leverage their backend databases, applications, and so on, in a mobile fashion. Doctors are using wireless networks to access patient records, submit prescriptions to the pharmacy, query databases, or even consult with other physicians. Traditionally, doctors and nurses would jot down patient information on a chart and enter it manually into the system at a later time. By enabling the nurse or doctor to enter that same information into a mobile device, the administrative burden is lessened significantly.

This process also decreases the amount of time a patient spends in the hospital. For example, a doctor might wish to discharge one of her patients at the start of her rounds. Without wireless capabilities, she might not enter the discharge information into the system until she completes her rounds two hours later; it could take another two hours to be accessed by the patient's nurse, who is away from her floor terminal. Had the doctor and nurse been equipped with wireless devices, the information would have been communicated immediately, the patient discharged and that bed filled by another patient. Not only did that four-hour time frame inconvenience the patient, but it also tied up a bed that could have been used for another patient. In short, more patients equal more revenue.

Financial

The ability to access real-time information is critical to the success of a financial institution. Stock markets are volatile, prices fluctuate and the success of a particular transaction is based on the speed with which it occurred. Traders on the stock exchange are now using wireless devices to update information in real time, rather than relying on frantic hand signals from the trading floors and scribbled receipts.

The New York Stock Exchange (NYSE) has installed a wireless backbone that has helped define the future of trading. Brokers are using PDA-like devices to send and receive orders, view market data, and send instant messages to other areas of the exchange. Wireless phones enable them to stay in constant contact with their partner firms, other brokers, or the wired trade terminal. What this means is that a transaction can now happen from anywhere on the floor without the broker being confined to a trading terminal or wired phone line. Furthermore, the transactions are immediate and the need for paper receipts is eliminated.

Academia

A growing number of colleges and universities are installing wireless networks in classrooms, libraries, and dormitories. This approach fosters greater student interaction, collaboration, and research opportunities within the classroom setting. Imagine the chaos if students were required to plug their laptops into a wired network several times a day from various points in the campus environment. The amount of cabling it would take just to accomplish this for one lecture hall could justify the ROI for installing wireless access points across the campus.

The wireless technology extends beyond the classroom setting. Students no longer need to compete for the limited resources available at the campus computer labs. Rather, they can access their data from just about any location on campus from the campus union, to the library and even their own dorm room.

Industrial/Factory

Inventory tracking, quality assurance, and material management are just a few examples of how wireless technologies have evolved in the industrial environment. Laptops can be mounted on forklifts in a distribution center to communicate to the workers that products need to be loaded onto trailers. Inventory can be tracked via wireless scanners as it moves through an assembly line. Giving engineers the ability to track changes, production issues, or faulty processes with handheld devices rather than pen and paper enhances quality assurance.

Retail and Restaurants

One of the primary applications used in the restaurant industry is for food orders being transmitted to the kitchen via a handheld device. Customer service is also enhanced by the usage of pen-based tablets by the hostess to track open tables and how many people are currently on the waiting list. The goal is to move people through the restaurant quickly while providing the best customer service possible.

Retail chains are migrating toward wireless networks to extend their point-of-sale (POS) solutions. Companies can add more cash registers and communicate the POS data to a central location without having to run cable to several locations throughout the store.

Wireless Internet Service Providers

In addition to benefiting traditional vertical markets, wireless has encouraged the growth of new vertical markets as well, including Wireless Internet Service Providers and Wireless Hotspot Providers. Both provide wireless Internet access to enterprises and consumers, directed at consumer or business locations, or supplying access to locations where wireless is desirable such as airports or coffee houses. While these particular markets have suffered from intermittent growth and downsizing, it's clear that wireless is an innovative market that will continue to accommodate growth in the future.

Bluetooth

The student will have a basic understanding of how the Bluetooth and Zigbee protocols work and the security issues that surround them.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Bluetooth

This section intentionally left blank.

Bluetooth

- Used to connect disparate devices
 - Laptops, PDAs, cell phones, headsets, and printers
- No line-of-sight requirement
- Supports data, voice, and content-centric applications with Bluetooth profiles
 - Cable replacement, not just networking
- Up to seven simultaneous connections
- Over several billion Bluetooth devices in 2014 and continued growth in 2015

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Bluetooth

Bluetooth was first announced in 1998. It promised to be an affordable, low-power wireless solution that would be attractive as a cable-replacement technology. Using Bluetooth, vendors would no longer need to ship unique networking cables with their products, relying on an inexpensive, integrated wireless card for all connectivity needs.

Using Bluetooth, consumers can connect any type of supported wireless device, from cell phones to PDAs, to talk with each other using a common technology. For example, you could use your cell phone to pull phone numbers from your PDA, or you could send documents from your laptop to a remote printer without physically being connected. Many wireless headsets for cell phones have adopted Bluetooth for wireless connectivity between an earpiece and the cell phone.

Unlike infrared networks that had limited success as a cable-replacement technology, Bluetooth has no line-of-sight requirements, making it much more flexible and user-friendly. Supporting data, voice, and content-centric applications such as streaming video or other data sources, Bluetooth networks meet a wide range of functionality requirements making it attractive to different connectivity needs. Capable of supporting up to seven simultaneous connections, a single Bluetooth adapter can communicate with several nearby devices simultaneously without being forced to connect and disconnect from different networks. With several billion Bluetooth devices being deployed in 2014, organizations will continue to see growth in Bluetooth devices.

Bluetooth Specification

- Range: ~1m, 10m, 100m
- Maximum bandwidth: 2.1 Mbps (EDR)
- Frequency: 2.4 GHz, FHSS
 - High degree of interference immunity
- Planned usage to replace all cables with peripheral computing
- Price goal: \$5 per radio unit

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Bluetooth Specification

The Bluetooth specification accommodates three classes of devices to differentiate the supported ranges of each device:

- Class 3 devices were available with the first deployment of Bluetooth hardware, supporting a range of approximately 1 meter. Class 3 devices are typically found in only early Bluetooth deployments or older hardware.
- Class 2 devices quickly followed, supporting a range of 10 meters. Class 2 devices are the most common hardware deployed for Bluetooth, commonly found in phones, wireless headsets and laptop computers.
- Class 1 devices are a more recent hardware option, supporting Bluetooth connectivity up to 100 meters. Primarily available in the form of USB dongles for laptop and desktop computers, the range of class 1 devices rivals that of 802.11 wireless networks.

Bandwidth in Bluetooth networks has not been a tremendous concern, because Bluetooth was not designed as a high-speed network topology as this is unnecessary for most Bluetooth applications. The most recent Bluetooth specification introduced support for rates up to 2.1 Mbps (improved from the previous support for approximately 1 Mbps) using the Enhanced Data Rate Bluetooth specification (EDR).

Bluetooth networking is based on a radio technology known as Frequency Hopping Spread Spectrum (FHSS). Using FHSS, Bluetooth networks rapidly hop between 79 different channels in the 2.4-GHz band, transmitting and listening in turn with other devices on the network. This process is seamless to the end user, but provides a high degree of interference immunity, limiting the ability of malicious or accidental interference on a single channel from affecting all communication.

With a pricing goal of \$5 per Bluetooth implementation (including radio interface, software and system memory requirements), Bluetooth is well-positioned as a mechanism to replace all peripheral computing cables in use today.

Bluetooth Security

- End user utilizes a PIN between multiple devices; it's 4-16 characters in length
- Bluetooth uses the pin and its MAC address to generate security keys
- Keys are used to authenticate Bluetooth "peers" and to encrypt transmission data
- Some devices must use fixed PINs
- Sniffing risk when devices first pair

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Bluetooth Security

Bluetooth security provides a simple (from the end-user perspective) means of authentication and data encryption when communicating between devices. Authentication starts with a user selecting a personal identification number (PIN) to authenticate other devices in the Bluetooth piconet ("small network"). The PIN is then entered manually into each device that needs to communicate with other Bluetooth-capable devices.

Bluetooth security relies on the secrecy PIN selected by the user and the MAC address of the Bluetooth device, known as a BD_ADDR (pronounced "bee dee adder"). When two devices connect for the first time, they use the PIN and BD_ADDR information to generate permanent link keys that are stored on the each device. Subsequent communication between the devices does not require the PIN input from the user again, instead relying on the stored link keys for authentication and encryption.

In many cases however, Bluetooth devices have security requirements or a desire for a secure operating environment without the ability to uniquely select a PIN. This is often found with Bluetooth Headset devices that do not have a human-to-machine interface (HMI). The lack of a HMI results in a fixed PIN selection for the headset, commonly "0000" or "1234."

Because the PIN is needed only at the initial pairing of devices, the risk of a static PIN is a target for an attacker when the devices initially pair. The Bluetooth Special Interest Group (SIG) that certifies and specifies how Bluetooth networks should interoperate initially dismissed this issue by recommending customers pair devices only in a trusted environment, free from the threat of unknown attackers sniffing the wireless medium. However, recent research published by the University of Tel Aviv, Israel by Yaniv Shaked and Avishai Wool presented findings that could be exploited by an attacker to force Bluetooth devices to re-pair in a hostile environment in order to mount an attack against the Bluetooth PIN.

Bluetooth Security Issues

- Susceptible to eavesdropping
- Bluetooth's use of SAFER+ encryption has been shown to be weak
- Simple PIN numbers are often poorly selected and inadequate security
- Tools such as RedFang and BlueSniff are designed to locate Bluetooth networks
- Bluetooth PAN APs can expose wired networks

Configuring devices in non-discoverable mode will help, but not ultimately protect Bluetooth networks

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Bluetooth Security Issues

Like all wireless communication mechanisms, Bluetooth networks are susceptible to eavesdropping attacks where an attacker can passively or actively monitor communication between devices on the Bluetooth piconet. The use of FHSS networking makes passive monitoring on Bluetooth networks a more complex attack for an adversary, but it is only through obscurity that an attacker with the right equipment (a Bluetooth protocol analyzer) can passively capture and record all Bluetooth network activity within range.

The Bluetooth algorithm used for encryption and authentication is based on the SAFER+ cipher by Cylink. SAFER+ was one of the submission candidates for the AES algorithm to be used by the U.S. government, but was rejected due to performance limitations with the reference implementation, and due to a weakness that reduced the effective key length when used with 256-bit keys. In the Bluetooth implementation of SAFER+, 128-bit keys are used for authentication and encryption, but are based on the relatively weak input of a PIN. If an attacker is able to capture the pairing process with a Bluetooth sniffer, it is possible to mount a brute-force attack against the PIN, recover a 4-digit PIN in as little as 63 msec on a Pentium 4 3-GHz system. What's more, many Bluetooth users will select weak PINs such as "0000" or "1234," further simplifying the attack.

In the past several years, security researchers are picking up an increased level of interest in Bluetooth PANs.

Tools that can be used by an attacker to locate and circumvent the security of Bluetooth networks include "RedFang" and "BlueSniff". Armed with these tools, it is possible for an attacker to locate and attack Bluetooth networks. Combined with the ability to correctly guess or attack the selection of a PIN, it's also possible for an attacker to monitor telephone conversations using a Bluetooth headset, decrypt data exchanged between a laptop and a PDA, or intercept keyboard and mouse commands from Bluetooth-enabled devices. What's more, some Bluetooth devices can be configured to extend access to the wired network over a wireless connection, similar to an 802.11 wireless network, with the Bluetooth Network Encapsulation Protocol (BNEP).

In response to Bluetooth security vulnerabilities, many vendors will recommend to customers that they configure their Bluetooth devices in "non-discoverable" mode after initially pairing with other Bluetooth devices. This will prevent a Bluetooth device from being casually discovered, but does not ultimately protect Bluetooth devices from a determined attacker. Because a device in non-discoverable mode must still respond to a PAGE request from another Bluetooth device, it is possible to scan all possible Bluetooth MAC addresses (BD_ADDR) for a given range to identify a device in non-discoverable mode. Further, if an attacker has access to a Bluetooth sniffer device, he can get access to raw RF signals and identify the presence of Bluetooth networks by examining packet traces.

Even if non-discoverable mode doesn't ultimately protect Bluetooth devices, it will often deter a casual attacker. The recent interest in Bluetooth security has produced new tools designed to probe and discover devices.

Bluesnarf Attacks

- Many Bluetooth vulnerabilities are in the Application Layer
- Bugs in several phones allow retrieval of phonebook and calendar
- Can also be used to make calls remotely, billed to victim
 - # bluesnarfer -r 1-10 -b 00:02:EE:3C:67:1F
 - device name: Nokia 6310i
 - + 1 - Joshua Smith : 3437464
 - + 2 - Mike Kershaw : 8675309
 - + 3 - Home : 7075206
 - + 4 - Max Moser/CEO : 5307071337

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Bluesnarf Attacks

One Bluetooth attack that has received a lot of attention is known as "Bluesnarfing." This and other similar attacks exploit Bluetooth weaknesses at the Application Layer, connecting to a discoverable device without providing security credentials and remotely accessing the device.

In a Bluesnarfing attack, the victim device is remotely accessed by the attacker to create a virtual serial port connection. This allows the attacker to remotely enter AT commands as if they were connected to the phone using a local serial cable. With the ability to enter AT commands, the attacker can download and change sensitive information from the phone, including:

- Phonebook entries
- Calendar entries
- Mobile provider IMEI information (uniquely identifies a GSM or UMTS mobile phone)
- Set up call forwarding
- Initiate calls to arbitrary phone numbers

The example in this slide demonstrates the use of one implementation of the Bluesnarf attack, exploiting the popular Nokia 6310i mobile phone to download a list of contact book entries from the phone. A list of known phones that are vulnerable to the Bluesnarf attack is maintained at <http://www.thebunker.net/security/bluetooth.htm>.

Protecting Bluetooth

- Configure devices in non-discoverable mode
- Audit the environment for Bluetooth devices
- Whenever possible, use a strong PIN that is at least 12 characters
- Pair devices only in a trusted environment
- Encourage vendors to implement SIG 2.0 specification/PKI support

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Protecting Bluetooth

In order to protect the privacy and confidentiality of Bluetooth networks, it is recommended users configure their devices in non-discoverable mode after completing the initial pairing process. Ultimately, this will not defeat a determined attacker, but will likely mitigate the majority of casual attacks that exploit vulnerable Bluetooth devices.

In order to enforce a policy that all Bluetooth devices must be used in non-discoverable mode, organizations can use tools like BlueScanner for Windows or Bluesniff for Linux.

Using scanning tools, administrators can manually identify and assess Bluetooth devices in use throughout their organizations by walking through their facilities while scanning for devices. At the time of this writing, there are no "fixed scanning" options available that are capable of distributed, real-time Bluetooth scanning support.

Whenever possible, Bluetooth users are encouraged to use a strong PIN consisting of at least 12 characters in length. PIN values that are 8 characters or shorter have been shown to be extremely weak, because an attacker can determine the PIN using a brute-force attack in less than 13 hours. PIN values that are 4 digits in length can be recovered in near real time.

When initially pairing devices, it is recommended that pairing happen in a secure environment that is at least likely to be free from an attacker passively capturing network traffic. For example, if a PDA and laptop need to be paired to exchange data over Bluetooth, pair the two devices in a secure office environment before connecting the two devices in a public location such as a coffee house or conference venue. This limits the risk of an attacker being able to capture enough information to mount a brute-force PIN attack.

Finally, the Bluetooth specification accommodates a stronger level of encryption support, utilizing public key cryptography that defeats many of the attacks focusing on PIN selection. However, few vendors have adopted this enhanced security model. Users are encouraged to communicate their desire for a more secure Bluetooth operating environment to their product vendors.

ZigBee Wireless

- Based on 802.15.4 specification
- Similar to Bluetooth as lost-cost, cable-replacement technology
- Targets product tracking, medical, and industrial sensor/control networks
 - Honeywell to deploy ZigBee in HVAC systems for management
- Close to 100 million nodes in 2014 and significant growth expected in 2015

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

ZigBee Wireless

ZigBee is another emerging wireless technology, based on the IEEE 802.15.4 specification. Similar to Bluetooth technology, ZigBee is a competing wireless specification designed at replacing cables, but targets specific vertical markets instead of general consumers. While Bluetooth is attractive as a general-purpose cable replacement technology, ZigBee is designed for use in product tracking, medical device monitoring, industrial sensor monitoring, control networks, and home automation systems. Unlike Bluetooth, ZigBee is designed to be a simple protocol implementation, requiring fewer memory and processor resources to deploy ZigBee technology (a complete ZigBee implementation is distributed by some vendors on a single 128 Kbyte memory card).

One example of a ZigBee deployment has been publicized by Honeywell International, embedding ZigBee radios in HVAC systems. With ZigBee wireless support, an administrator can connect to the HVAC device with a ZigBee client card to manage and monitor maintenance history, utilization, and control information.

ZigBee Specification

- Range: 10-75 meters
- Frequency: 868 MHz, 915 MHz, 2.4 GHz, DSSS
- Rate: 250 Kb/s @ 2.4 GHz; 40 Kb/s @ 915 MHz; 20 Kb/s @ 868 MHz
- Focuses on low-power consumption
 - Goal of 10 years service on 1 battery
- Planned usage to replace home wiring systems for monitoring, control
- Price goal: \$4-6 per radio unit

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

ZigBee Specification

The design of the ZigBee specification was geared to accommodate very inexpensive radio components with minimal software and memory requirements to implement the ZigBee stack. Ranges on ZigBee networks are between 10-75 meters, depending on the hardware selection and radio frequency in use. Unlike Bluetooth that supports only the 2.4-GHz wireless band, ZigBee devices can transmit in 2.4-GHz, 915-MHz, and 868-MHz bands. Higher-frequency bands accommodate faster data speeds, but lower-frequency bands accommodate greater transmission ranges. ZigBee also operates with Direct Sequence Spread Spectrum modulation, unlike the Frequency Hopping modulation used by Bluetooth networks.

In order for ZigBee to be a desirable technology for control and monitoring systems, the devices must accommodate an extended-length battery system. The ZigBee radio cannot assume to draw power from the device it is interacting with because the radio may be used to control power input or operate when a device is not powered on.

ZigBee accommodates sustained battery life by limiting the amount of time the radio is used. In the case of a lighting system with ZigBee support, the system may remain inactive until it receives a command to enable light, at which time the radio turns on, transmits the signal to turn on the light at the remote system, waits for a status acknowledgement message, and then returns to a sleep state until another command is received.

The design goal for ZigBee is to accommodate 10 years of sustained wireless connectivity on a single battery, but most vendors have marketed their ZigBee products as supporting 3-5 years battery life on one or two alkaline batteries.

Planned deployments for ZigBee include the replacement of cabling for home wiring systems. Commonly known as "home automation" or "domotics" (the application of automated technologies to domestic appliances), ZigBee devices are available to monitor all reporting systems in the house such as fire and smoke alarms, electricity and water utilization and alarm systems interacting with dial-operated systems to notify fire and

police departments in the event of an emergency. Some systems are sold with a potential cost-savings benefit to consumers, reducing the cost of utility bills by closely monitoring temperature sensors and appropriately controlling climate controls as well as curtains, shades, and blinds without human interaction.

With a pricing goal of \$4-\$6 dollars per ZigBee radio, it is likely that an increasing number of manufacturers will adopt ZigBee technology to accommodate integration with other ZigBee-compliant products.

ZigBee Security

- It can accommodate security at MAC, Network, and Application layers
- It relies on master keys set by manufacturer, installer, or end user
 - Generates link keys to encrypt traffic
- Encryption-based on AES-CCM
- Security optional; AES may be too resource-intensive for lightweight devices
 - Balance between battery life and security

Organizations must evaluate implementation security.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

ZigBee Security

The ZigBee specification has a section dedicated to the security of ZigBee networks, accommodating security at the MAC, Network, and Application Layers. This allows ZigBee application developers to have extra flexibility in where they implement security functions, relying on security at the MAC Layer or at the upper-layer network and application functions. For simplicity, however, the ZigBee specification requires that the same key be used for all three layers of security, implying that if an attacker has compromised the MAC Layer link key, he will also be able to decrypt traffic at the Network Layer and Application Layer using the same key.

The selection of a key is done at installation time, and is set by the manufacturer, installer, or end user. After specifying an initial "master" key, two ZigBee devices will establish a mutual link key that is used to authorize other ZigBee nodes and to encrypt and decrypt traffic. This is beneficial to the end user, as a compromised link key will reveal only the data between two nodes; it does not also reveal the secrecy of the master key and link keys used between other ZigBee devices.

The encryption algorithm for ZigBee networks is based on the recently adopted American Encryption Standard (AES) Rijndael cipher in CCM mode. CCM stands for Counter with CBC-MAC; CBC-MAC stands for Cipher-Block-Chaining, Message Authentication Code.

Rijndael is a respected cipher due to its selection by the U.S. government as the replacement for the former Digital Encryption Standard encryption algorithm. Combined with CCM, Rijndael also provides integrity protection that prevents encrypted traffic from being modified during transmit.

While AES is a strong cipher that is suitable for many different encryption needs, it is also a resource-intensive protocol, which may be too much of a burden for lightweight ZigBee devices to accommodate. For this reason, the ZigBee specification has made the use of AES and encryption *optional*. This allows organizations to sell ZigBee products with the ZigBee seal of approval, without necessarily disclosing whether or not the product supports the ZigBee security mechanisms. Organizations should work closely with vendors to identify what security options are available with their ZigBee products to ensure they are taking advantage of the confidentiality and integrity protection mechanisms available.

802.11

The student will be able to identify the different 802.11 protocols and understand key characteristics.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

802.11

This section intentionally left blank.

IEEE 802.11 Wireless

- Supports ad-hoc and infrastructure networks
- Supports roaming, fragmentation, and reliable data delivery (positive acknowledgement)
- Branched into:
 - 802.11b supports up to 11 Mbps @ 2.4 GHz
 - 802.11a supports up to 54 Mbps @ 5 GHz
 - 802.11g supports 22/54 Mbps @ 2.4 GHz
 - 802.11n supports 100+ Mbps @ 5 GHz

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IEEE 802.11 Wireless

The 802.11 standard was approved in 1997 by the IEEE 802 Committee. This standard has several key elements that make it the most widely adopted wireless LAN standard in use today:

- Supports ad-hoc and infrastructure networks.
- Accommodates roaming between multiple access points without losing connectivity.
- Supports large data packets through the use of fragmentation at Layer 2.
- Provides reliable data delivery when experiencing interference due to a requirement to positively acknowledge all data traffic received from an access point or wireless station.
- Builds on existing standards for data encapsulation (802.2 LLC).
- Power conservation techniques extend the battery life of wireless devices.
- Albeit weak, the IEEE 802.11 specification included a method for encrypting data using a shared secret and the RC4 encryption algorithm. Known as wired equivalent privacy (WEP), this algorithm was the first IEEE standard to perform encryption and authentication at the Data Link Layer.

The initial 802.11 specification branched into multiple specifications that utilize varying technology for frequency modulation that would support varying data rates:

- *IEEE 802.11b 1999*: Supports a theoretical throughput of 11Mbps in the 2.4-GHz spectrum. Actual throughput for 802.11b networks is commonly closer to 6Mbps. The 802.11b specification is widely adopted and is the most popular wireless LAN protocol in use today.
- *IEEE 802.11a 1999*: Supports a theoretical rate of 54Mbps. Using the 5-GHz spectrum, the 802.11a specification was not widely adopted until 2003.

- *IEEE 802.11g 2003*: Supports a theoretical rate of 56Mbps using the same frequency as 802.11b. Designed to boost the speed of networks running in the 2.4-GHz band, 802.11g suffers from many drawbacks in speed and performance when used in conjunction with 802.11b devices.
- *IEEE 802.11n (~2009)*: Supports actual throughput of 108 Mbps to 320 Mbps and will drastically improve performance in both the 2.4-GHz and 5-GHz bands. The "n" standard is focused not only on increasing raw data speed, but on significantly reducing the amount of management overhead that robs the other standards of realized throughput.

The IEEE wireless LANs specifications are available at <http://standards.ieee.org/getieee802/802.11.html>.

IEEE 802.11i, 802.1x, EAP

- 802.11i provides strong encryption, replay protection, and integrity protection
- 802.1x provides network authentication
- EAP types specify how authentication is protected
- Different EAP types are suitable for different environments:
 - Consider clients, directory type, hardware

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC.

IEEE 802.11i, 802.1x, EAP

In order to address the failures with the WEP encryption protocol, the IEEE started working on a new encryption standard for 802.11 wireless networks in 1998. Ratified on June 24, 2004, the 802.11i specification accommodated two replacement encryption mechanisms for WEP, one that could be retrofit into existing hardware, and a second design that would be a "completely secure" solution, requiring new hardware for implementation. Known as the Temporal Key Integrity Protocol (TKIP) and the Counter-Mode/CBC-MAC Protocol (CCMP), respectively, these algorithms represent a significantly more secure option for organizations to deploy wireless LANs. Both protocols protect information on the wireless network through strong encryption, replay protection, and integrity protection.

While 802.11i accommodates privacy and encryption for network traffic, it does not address the issue of authentication. This was the task of the IEEE 802.1x working group, which designed a framework for network authentication. This authentication framework accommodates several different authentication protocols known as Extensible Authentication Protocol (EAP) types.

Several different options are available for organizations when selecting an EAP type. The correct EAP type for your organization is related to the client operating systems that are in use, the back-end authentication systems (Windows Active Directory, LDAP, RADIUS, etc) and the wireless access points and wireless cards in use.

WEP Security Issues

- WEP has proven to be an insecure encryption mechanism
- Shared secrets don't stay secret
- Inability to rotate WEP keys produces stagnant shared secret implementations
- Flaws in WEP implementation permit recovery of shared secrets
- Accelerated WEP cracking defeats dynamic WEP

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

WEP Security Issues

Several weaknesses in attempts to secure 802.11 networks have made it difficult for administrators to keep their wireless networks safe. Attackers armed with the knowledge of common vulnerabilities in 802.11 networks are readily attacking 802.11 networks. These attacks are being launched to grant unauthorized access to wireless networks, to perform denial-of-service attacks, and to collect sensitive information from private networks.

The wired equivalent privacy (WEP) algorithm was included in the IEEE 802.11 and later specifications to provide data confidentiality on wireless LANs. The 802.11-1997 specification introduced WEP as an implementation of the RC4 encryption protocol, the same encryption protocol used by the SSL and TLS protocols. WEP is based on a pre-shared secret that is common to all stations that participate in the same wireless network. In this implementation, an administrator would visit workstations that wanted to communicate on the wireless network and manually configure them with the same shared secret. The access points and clients would use the shared secrets to encrypt and decrypt data that was sent on the wireless network.

The 802.11 specification never included rotating the shared secrets on a wireless network, so many networks continued to use the same shared secret for all of their wireless clients. The inability to easily change the WEP keys on all workstations became an even more daunting problem as the utilization of wireless networks continued to grow and more people depended on the shared secret configured on their workstation.

Unfortunately, WEP proved to have another insufferable flaw in its implementation; Fluhrer, Mantin, and Shamir first reported this flaw in their paper, *Weaknesses in the Key Scheduling Algorithm of RC4*. Their paper identified a method by which an attacker could recover the shared secret from nothing more than the encrypted data collected from a wireless network. Shortly after the paper was published, tools were released that would recover the secret WEP key used on a network after collecting millions of packets from a wireless network. The process of recovering a WEP key in this fashion commonly took days on a network that had little or moderate traffic levels.

Tools used by attackers to recover shared WEP keys on a network include WEPCrack, AirSnort, and dwepcrack. These tools are written for Linux or BSD systems. Although quite effective for attacking wireless LANs utilizing the WEP algorithm, they required a dedicated attacker with patience to recover the shared secret WEP key from a wireless network.

Recent attack tools against WEP have been developed to make the process of recovering WEP keys and attacking wireless networks even faster. Tools such as wnet/reinj and WEPWedgie accelerate the process of collecting packets from a wireless network, often resulting in an attacker's ability to recover a shared secret from a network using WEP in one hour or less.

Recognizing that the WEP algorithm was an insufficient method of protecting wireless networks, the IEEE and IETF have developed alternate solutions to protect wireless LANs; however, many organizations still use WEP technology due to limitations with legacy hardware and because of the cost of replacing all hardware to accommodate stronger encryption mechanisms.

Wi-Fi Protected Access

- Wi-Fi Alliance performs interoperability testing for 802.11 hardware vendors and consumers
- It enables early adoption of improved security
- WPA is an improvement over WEP on old hardware (TKIP)
- WPA2 is a vast improvement over WEP; it requires AP and NIC replacement (AES-CCMP)

**Set organizational purchasing policy
to require WPA2 interoperability
for new wireless purchases.**

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Wi-Fi Protected Access

The Wi-Fi Protected Access (WPA) specification was adopted by the Wi-Fi Alliance before the IEEE 802.11i specification was completed to give organizations an opportunity to improve the security of wireless networks. In 2003, many organizations were becoming increasingly concerned about the security of wireless networks, without a clear solution from the IEEE to replace WEP. Although the IEEE 802.11i committee had formalized a replacement for WEP, the 802.11i specification was otherwise incomplete.

The Wi-Fi Alliance adopted the 2003 draft of the 802.11i specification and started performing interoperability testing for vendors using the Temporal Key Integrity Protocol (TKIP). This testing process certified a vendor product as WPA-compliant, focusing on the implementation of TKIP as a mechanism to replace WEP on existing hardware. After the 802.11i specification was ratified in June 2004, the Wi-Fi Alliance also adopted the AES-CCMP cipher mechanism designed for new hardware. The testing process for compliance with TKIP and AES-CCMP became known as WPA2.

Organizations are encouraged to adopt the TKIP and AES-CCMP encryption mechanisms to improve the security of their 802.11 networks. Organizations can adopt TKIP with most existing hardware.

Wireless Security

The student will have a basic understanding of the misconceptions and risks of wireless networks and how to secure them.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Wireless Security

This section intentionally left blank.

General Misconceptions

- "I don't need to worry about security; we aren't using wireless for sensitive data."
- "We don't have any wireless."

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

General Misconceptions

"I don't need to worry about security; we aren't using wireless for sensitive data."

This is probably the most common misconception about wireless LAN security. Many organizations have installed wireless LANs for collaboration in small workgroups, for limited deployment to meet specific job functions, or simply to experiment with the technology. Typically deployed with their factory default configuration settings, these networks are often the most vulnerable. It is also common for an organization to simply forget about the "temporary" wireless LANs they install, or to forego security measures because they believe the wireless LANs are not being used for business-critical functions.

The critical vulnerability with these wireless LAN installations is that they are completely open for an attacker to utilize for his own nefarious purposes. Attackers leverage these vulnerable networks to their advantage, using them to deliver SPAM e-mail messages, share copyright-infringed software, music, and movies, or gain anonymity when launching attacks against other organizations' networks. The lesson here is that no wireless networks should be deployed without first completely understanding the risks to the business in the event that they are exploited.

"We don't have any wireless."

Some organizations have adopted policies stating that wireless networks will not be used for their organization. Although this might be a good decision for some organizations, it is not a useful policy unless it can be enforced. Enforcing such a policy can be a time-consuming and difficult task for many organizations.

Organizations that believe they do not have any wireless networks connected to their corporate network often discover that they have unauthorized wireless access points, which are typically deployed with little or no security. Often deployed with innocent intentions, unauthorized wireless networks make their way into organizations, opening up access to the internal network and bypassing perimeter defense systems.

Another common implementation of unauthorized wireless networks is found in laptop and handheld computers that come bundled with built-in wireless cards. This built-in hardware rarely offers the opportunity to be disabled in hardware and automatically connects to available wireless networks in default configurations of Windows XP and Windows Mobile operating systems. This often presents an attacker with the opportunity to connect to vulnerable computers without being detected by wired-side intrusion detection systems.

A final vulnerability to consider is the use of wireless LANs by users who connect to corporate networks from home over VPN. A home user is likely to have minimal security measures (if any) on their wireless network. A computer connected to both a corporate VPN and a home wireless LAN gives an attacker the opportunity to compromise a vulnerable host and utilize the existing VPN connection to gain access to the corporate network.

The lesson for this misconception is that, even if they are not deploying wireless networks or they have a policy against the use of wireless, all organizations must consider their vulnerabilities in relationship to wireless networking.

Technical Misconceptions

- "We cloak our SSID, so people can't join our wireless network."
- "We filter weak IVs, so WEP is safe."
- "MAC-based access control restricts access to authorized users."
- "Technology XYZ by itself protects us."

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Technical Misconceptions

"We cloak our SSID, so people can't join our wireless network."

Although most access points offer the ability to hide the service set identifier (SSID) or network name of the wireless LAN, this feature should not be treated as a method of preventing unauthorized access. Although an attacker must have the SSID to join the wireless network, he can harvest this information even if an access point is configured not to advertise its network name. By passively listening to the network, an attacker can wait for a valid client to associate to the wireless network and capture the network name information. The lesson for this misconception is that the SSID should not be treated as if it were a password or a means of access control to the wireless network.

"We filter weak IVs, so WEP is safe."

To mitigate the vulnerabilities described in the Fluher, Mantin, and Shamir paper on weaknesses in the RC4 protocol, some vendors have made updated software available to customers to mitigate the ability of tools such as AirSnort and WEPCrack from being able to recover WEP keys. Although it effectively prevents these tools from recovering the shared secret used for WEP, weak IV filtering does not help administrators manage the management problems associated with using WEP to protect wireless networks. Specifically, administrators must still manage the distribution of shared secrets on all computers that participate in the wireless network without being able to easily update and rotate the WEP keys.

Weak IV filtering also does not prevent other attacks against WEP, including an attacker's ability to inject arbitrary packets into a WEP network without knowing the WEP keys. With filtering used to accelerate the WEP-cracking process, an attacker can utilize this weakness to port-scan, discover, and exploit hosts behind a wireless network—all without knowing the WEP key in use.

The lesson for this misconception is that, despite efforts to improve weaknesses in the protocol, the WEP protocol cannot be used as a mechanism to secure wireless networks.

"MAC-based access control restricts access to authorized users."

Nearly all access points offer the ability to maintain a list of MAC addresses that are allowed to connect to the wireless LAN. The MAC addresses on wireless cards are associated with specific users and used as a means of restricting access to the wireless network.

Unfortunately, this means of access control can be circumvented by an attacker. Each packet sent from a legitimate workstation identifies the source MAC address that sent the packet. An attacker who discovers a network restricting access based on MAC addresses can simply monitor the network to identify people who are sending traffic on the network. The attacker can identify valid MAC addresses that are permitted to use the wireless network, regardless of encryption protocols in use on the network. With a list of authorized clients, the attacker can simply select an authorized MAC address and change his wireless card to utilize the same MAC. After the attacker updates his computer with an authorized MAC address, he can communicate on the network as if he were an authorized user.

The lesson for this misconception is that attackers can easily circumvent MAC-based access control, which is not a viable means of access control for wireless networks.

"Technology XYZ by itself protects us."

No single technology can sufficiently protect wireless networks. Administrators wishing to protect their networks should consider deploying many security layers, including strong encryption methods, distributed firewalls, and intrusion detection systems between wireless and wired networks, personal firewalls on wireless clients, and authentication, authorization, and accounting services.

Risk Misconceptions

- "DoS attacks require expensive hardware that is not easily accessible."
- "Segregating our wireless LAN eliminates our risk of exposure."
- "This whole wireless thing is secure by default, right?"

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Risk Misconceptions

"Denial-of-service attacks require expensive hardware that is not easily accessible."

Unfortunately, this statement could not be more incorrect. Denial-of-service (DoS) attacks against wireless networks are easy to implement due to weaknesses in the 802.11 specification and are impossible to prevent with current technology. With a \$10 wireless card and readily-available software, an attacker can launch DoS attacks against victim networks that completely disable all access to the wireless LAN. They can launch these attacks with breadth and target them against an entire building or office park, or against a single wireless LAN—potentially from a distance of several miles.

The lesson for this misconception is that DoS attacks are a real threat to organizations, and administrators have little opportunity to limit their exposure to attack.

"Segregating our wireless LAN eliminates our risk of exposure."

A common method for securing the implementation of wireless LANs is to segregate the wireless LAN away from internal networks with a firewall. Although it is a strong component of an overall wireless LAN security plan, this implementation still poses significant risks to an enterprise network.

Segregated networks typically open access only to those applications that are required for wireless clients or VPN systems that require authentication and strong encryption to access internal systems.

An attacker wishing to circumvent the security of these installations has several attack options. Application-level flaws can be vulnerable to attack on exposed servers, or an attacker can attempt to exploit vulnerable clients to access established VPN tunnels and thus, internal networks.

The lesson for this misconception is that while segregating wireless LANs away from production networks is a critical component of securing the enterprise, a defense-in-depth approach must be applied to adequately defend against attack.

"This whole wireless thing is secure by default, right?"

It is important for all levels of an organization—from the wireless LAN technicians up to the Chief Security Officer—to understand the risks associated with deploying wireless networks. Although the CSO probably does not need to understand the mechanics behind weaknesses in the WEP algorithm, it is important for them to understand and actualize these risks before authorizing the deployment of wireless networks. It is not uncommon for people to deploy wireless networks, treating them with the same security mechanisms they give to wired networks. Obviously, the deployment of wireless LANs is a completely different paradigm than traditional wired networks. When securing enterprise networks, communicating the risks, deficiencies, and advantages of wireless LAN deployment to all levels of an organization is clearly a critical part of a defense-in-depth strategy.

Top 4 Security Risks for WLANs

- Eavesdropping
- Masquerading
- Denial-of-Service (DoS)
- Rogue APs

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Top 4 Security Risks for WLANs

As the popularity of wireless networks increases, their inherent security flaws are receiving more and more attention. In recent years, wireless security (or lack thereof) has become the press's media darling.

Unfortunately, these reports are often incomplete or incorrect, and they often leave organizations with a false sense of security. This section focuses on the most critical security issues related to wireless networking:

- Eavesdropping
- Masquerading
- Denial-of-Service (DoS)
- Rogue APs

Eavesdropping

- Wireless transmissions do not obey property lines
- Anyone with a suitable receiver within range of the signal can eavesdrop
- Access to the network can be gained while being hundreds of feet away (for example, from a parking lot or nearby street)
- Distances can be increased with antennas
- Anyone can gain access to confidential information

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Eavesdropping

Eavesdropping is a trivial matter in a wireless RF environment. Data sent over the radio path can be intercepted by anyone equipped with a suitable device that happens to be listening on the same frequency. As if that was not bad enough, the devices needed to perform eavesdropping are inexpensive and very easy to use. But wait, it gets even worse! It is virtually impossible to detect a hacker listening in on your wireless communication.

As we learned earlier, wireless RF has the capability to travel beyond the confines of a building, and the signal can often be picked up for 300 feet or more beyond its intended recipient. Would-be attackers can eavesdrop on wireless networks from remote locations, such as a company parking lot or building lobby, and potentially gain access to confidential information.

An attacker wishing to eavesdrop on a wireless network likely wants to place as much distance as possible between the victim network and the attacker's location to avoid detection. To aid them in this venture, attackers employ range-extending antennas connected to their wireless cards. These antennas are sometimes commercial tools purchased over the Internet or in ham radio shops, or are home-brewed using nothing more than components that are available at Radio Shack and the local supermarket. Hackers have discovered that properly modified Pringles Chip cans, as well as other juice and soup cans, can extend the distance between an attacker and the victim network from 600 feet to several miles.

Eavesdropping Mitigation

- Use strong encryption in the lowest layer protocol possible
- Design your wireless networks with caution; minimize the coverage area
- Audit your network with a packet sniffer

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Eavesdropping Mitigation

Even if an attacker can hear a transmission, he cannot make sense of the information if the data is protected by encryption. It is important to use strong encryption methods that operate at the lowest possible layer of the OSI model. Consider the amount of information an attacker can glean from the Cisco Discovery Protocol (CDP), a protocol that is not encrypted when using IPSec or other encryption protocols that work at the Network Layer of the OSI model.

We have seen that the WEP protocol is inadequate for protecting wireless networks. Organizations should deploy stronger encryption protocols, such as TKIP (WPA), if they do not adopt WPA2, which uses AES encryption based on the final IEEE 802.11i amendment.

When designing wireless networks, it is possible to select range-limiting antennas to limit the exposure of information outside an organization's walls. Using alternate antennas, limiting signal output strength on radio cards, and placing wireless access points away from the exterior reaches of buildings will reduce the amount of wireless traffic that is sent outside a building's physical boundaries. Organizations should also consider working with their facilities' management departments to employ RF-limiting materials, such as wire-mesh installed in walls, ground-connected metal studs and beams, and even metal-additive in paint to limit RF leakage at the edges of buildings. Although it is not a singularly protective measure, limiting the range of wireless LANs makes it more difficult for an attacker to eavesdrop on vulnerable wireless networks.

Finally, organizations should audit wireless networks with a packet sniffer. LAN and security administrators can use either commercial wireless sniffers or open-source tools such as Kismet and Wireshark to eavesdrop on wireless networks and analyze the captured data. By eavesdropping on their wireless networks, administrators can identify vulnerable access points and understand the level of exposure to the organization.

Masquerading

- An attacker spoofs the identity of a legitimate node or AP
- Tricks unsuspecting users to giving up sensitive information
- Tricks an AP into authenticating malicious users
- "Evil Twin" attack is gaining popularity

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Masquerading

Masquerading is the term used to describe the activities of an attacker who impersonates the identity of legitimate nodes or access points in a wireless network. The attacker accomplishes this by spoofing identity information to impersonate an otherwise authorized client or access point. By making clients think that they are communicating with a legitimate access point, attackers can trick unsuspecting users into giving up sensitive information, trick access points into believing that they are authorized clients, or launch denial-of-service attacks.

Collecting Sensitive Information

By impersonating a legitimate access point, an attacker can offer network services to unsuspecting wireless users and try to trick them into giving up sensitive information such as usernames, passwords, or even credit card information.

Tricking an AP into Authenticating Malicious Users

Captive Web portals are a common means of authenticating wireless users without requiring an authentication client on workstations that need to access the wireless network. Commonly used for authenticating users at hot spots, for guest access to wireless networks, or at colleges that offer wireless access to students, a captive Web portal intercepts requests for a Web page and substitutes the page with a form that requests authentication. If a user enters authorized authentication credentials, he is granted access to resources beyond the Web portal system. To grant access to only legitimate systems, the captive Web portal system must keep track of authenticated and unauthenticated users. The Web portal system tracks the MAC addresses of authenticated clients to permit access to network resources. Systems that use MAC addresses that are not in the explicit permit list are denied access until authentication.

To bypass this method of access control, an attacker can simply masquerade as an authenticated client by changing his MAC address. Sometimes, when combined with a DoS attack against the impersonated system, the attacker changes his MAC address to a system that was actively communicating on the network. The captive Web portal system checks the traffic's MAC address and, unable to differentiate the attacker from the legitimate user, grants the attacker unrestricted access to the victim network.

Masquerading Mitigation

- Use mutual-authentication wireless protocols such as PEAP or TTLS
- Use SSL/TLS for passing sensitive information to web applications
- Educate users on the dangers of clicking "Yes" to digital certificate warnings

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Masquerading Mitigation

To protect against masquerading attacks, you must have some mechanism in place to authenticate users to an access point and authenticate the access point to the user. By requiring the access point to present authentication credentials to the user, it is possible to mitigate attacks such as those implemented in the AirSnarf tool. This is possible using the IEEE 802.1X network authentication protocol in conjunction with extensible authentication protocols that support mutual authentication, such as EAP/TLS, PEAP, and TTLS.

In many cases, implementing mutual authentication protocols is impractical. For instance, hot-spot locations cannot require that clients have 802.1X clients configured on their workstations and therefore must seek alternatives. SSL or TLS encryption protocols that utilize public-key infrastructure with digital certificates can be an alternative for hot-spot access and other Web-based applications. Using digital certificates to authenticate the Web server or captive Web portal system makes it much more difficult for an attacker to masquerade his identity as the legitimate network resource.

Unfortunately, many users have grown anesthetized by digital certificate warnings and simply click-through warnings generated by Web browsers warning of mismatched digital certificates. This gives the attacker the opportunity to bypass this security mechanism and impersonate the characteristics of the digital certificate in an attempt to collect private information. It is critical for system administrators to successfully implement SSL and TLS systems with current digital certificates and educate users about the potential dangers of ignoring invalid certificate warnings.

Denial-of-Service Attacks

- RF jamming techniques and tools are readily available
- Weaknesses in the 802.11 specification permit DoS attacks
- Bluetooth networks less susceptible; based on FHSS instead of DSSS/OFDM

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Denial-of-Service Attacks

Wireless networks are an easy target for denial-of-service attacks. Vulnerabilities in the 802.11 specification, flaws in the firmware of popular WiFi cards, and weaknesses in the nature of radio communications offer attackers opportunities to shut down wireless networks at their leisure.

RF Jamming Attacks

If an attacker has a powerful enough transceiver, he can generate so much radio interference that the targeted WLAN is unable to communicate effectively. Like eavesdropping, this kind of attack can be initiated from a distance. Although the attack is a bit more sophisticated than simple eavesdropping, the equipment needed is readily available, as well as instructions on how to carry out such an attack. Attackers can purchase RF-jamming equipment, which is designed to stress-test wireless frequencies to attack wireless networks. Alternatively, attackers can build their own RF-jamming tools using inexpensive hardware from popular electronics stores and plans that are available on the Internet. These tools are very effective at stopping all wireless activity, often covering all available channels in the 2.4-GHz or higher frequencies. These RF jamming attacks are specification-agnostic; they are equally effective against 802.11 and Bluetooth networks, as well as any other communication that uses the same frequency as the attacker.

Weaknesses in the 802.11 Specification

Using commodity wireless cards, attackers can masquerade their identity as legitimate stations or access points to launch DoS attacks. Because the 802.11 specification does not include any per-packet authentication mechanism, access points and stations do not have a way of verifying that each packet is indeed sent from its reported source address. Attackers utilize this weakness to send spoofed packets to victim clients on behalf of the access point, telling the victim to disconnect from the network. The victim station processes this packet as if the traffic was sent from the access point and disconnects from the wireless network. An attacker can send a sustained flood of these disconnect packets to the LAN broadcast address, thereby causing all stations to disconnect from the network and resulting in a sustained DoS attack.

Flaws in WiFi Card Firmware

All wireless cards rely on firmware that is bundled in non-volatile RAM to handle time-sensitive transmission functions. Administrators often overlook this firmware because it rarely requires upgrades and offers no configuration options. Recently uncovered vulnerabilities and flaws in WiFi card firmware have led to a more effective means of launching DoS attacks against 802.11 networks. By sending specifically malformed frames to stations that run flawed firmware, an attacker can produce several undesirable results, ranging from complete loss of network connectivity to crashing host operating systems. Fortunately, card vendors have begun recognizing these flaws and are offering patched firmware to resolve these vulnerabilities.

DoS Attack Mitigation

- Understand the impact of a DoS attack against your environment
- Deploy wireless intrusion detection systems
- Prepare a response strategy

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

DoS Attack Mitigation

Protecting against WLAN DoS attacks is difficult. Administrators can employ the same mitigation strategies as those described for protecting against masquerading attacks, including limiting the range of wireless networks and employing RF shielding in walls and windows. Unfortunately, these tactics are often inadequate for protecting networks from DoS attacks when an attacker is equipped with directional and high-gain antennas, such as those built from home-brew designs.

The best mitigation strategy for DoS attacks against wireless LANs is to clearly understand the impact of such an attack against your network and prepare an appropriate response strategy. What is the effect of a DoS attack against your production networks? In the event that you are under attack, what alternatives will you pursue to reestablish connectivity to mission-critical systems?

To quickly identify and assess the impact of DoS attacks, organizations should consider deploying wireless intrusion detection systems using commercial or open-source tools. Wireless IDSEs allow administrators to react quickly to attacks against their networks, and they might provide enough information to identify and locate attackers.

Rogue APs

- Unauthorized APs connected to a private network
- Often installed with default settings and no security
- Permits full access to a network for an unauthorized user
- Contributes to unauthorized information disclosure

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Rogue APs

A rogue access point (AP) is connected to a wired network without the authorization to provide wireless service to end users. Users who want wireless access or are unhappy with existing wireless services expose an organization's network by connecting an access point to the wired networks. These access points are commonly meant for home use and rarely offer anything beyond the most basic security settings. Attackers who identify these rogue access points can exploit the basic security settings and gain access to internal network resources. Administrators are often unaware of rogue access points on their network until they are discovered as part of a vulnerability assessment or system compromise analysis.

Rogue access points can also contribute to unauthorized information disclosure when attackers eavesdrop on these connections. An employee might decide to deploy a rogue access point in a conference room with the intention of enabling a workgroup to easily communicate and share documents. Although good intentioned, the user often does not realize the risk and exposure of such an installation and cannot detect an attacker who is harvesting all the shared documents from a parking lot or other off-site location.

Rogue AP Mitigation

- Perform rogue AP detection
- Use mutual authentication wireless protocols such as PEAP or TTLS
- Deploy 802.1x on your wired network
- Deploy wireless intrusion detection systems
- Deploy a strong wireless LAN

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Rogue AP Mitigation

Rogue access points with little or no security pose an obvious threat to any organization. Often organizations that have adopted "no wireless" policies are plagued with rogue access points due to the lack of any wireless access available to users.

To mitigate rogue access points, administrators should perform rogue AP detection using commercial or open-source tools like Kismet and Ethereal. This method requires an administrator equipped with a laptop or handheld device to walk through the hallways and offices of all the buildings that might be risks for rogue access points, to detect any unauthorized wireless activity. Unfortunately, this can be a difficult venture for large campuses that require alternate detection methods.

Scanning wired networks for characteristics that resemble wireless access points is an alternate, yet less reliable method of detecting rogue access points. Using vulnerability assessment tools such as Nessus, an administrator can scan all the nodes on the wired network to identify Web pages, login banners, and other characteristics to identify potential rogue access points. This method is useful in detecting users who are not trying to hide their activity with otherwise stealthy tactics, such as shutting off ICMP echo responses and disabling administrative interfaces that might be used to identify them.

Some organizations are planning on the deployment of 802.1X network authentication on their wired networks to mitigate (among several security issues) the threat of rogue access points. By requiring nodes to authenticate to the network before being granted access, administrators can prevent users from connecting access points to their production networks.

In lieu of manual rogue AP detection, organizations should consider deploying WLAN intrusion detection systems to constantly monitor their facilities for rogue access points. Some WLAN IDS systems even implement "rogue AP countermeasures," which use attacker-like denial-of-service attacks against discovered rogue access points to prevent anyone from connecting to the rogue until an administrator visits the site to remove the offending hardware.

Finally, organizations plagued with rogue APs should consider deploying wireless networks for their users. By taking control of wireless equipment deployment, administrators are in a much better position to set policy dictating how wireless LANs are used and to design and implement a preferred security solution for end users. Users are less likely to deploy their own rogue access points if a stable and reliable wireless LAN is available.

Steps to Planning a Secure WLAN

- Consider design at all layers of the OSI model
- Identify specific areas for coverage
- Maintain consistency in deployment
- Audit the WLAN for rogues and unauthorized clients
- Consider wireless IDS

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Steps to Planning a Secure WLAN

The key to protecting any wireless network is to employ a layered defense. There isn't any one technology that will stop all attackers, but multiple layers of protection will be more likely to defend wireless systems.

When designing wireless networks, consider security mechanisms at all layers of the OSI model. We have suggested using careful consideration when deploying wireless access points to limit the RF coverage at layer 1 to mitigate eavesdropping and denial-of-service attacks, and using strong encryption algorithms to protect data confidentiality at layer 2. We can continue this process to carefully evaluate and deploy appropriate security mechanisms, such as firewalls at layer 3 and 4, strong upper-layer protocols at layer 5 and 6 that support non-repudiation of data, and application-layer defenses such as hardened systems and applications at layer 7.

Deploying wireless LAN equipment in a consistent manner under a formal change-control process will reduce the probability of attack due to misconfigured equipment. It is not uncommon to discover an access point that has default SNMP community strings, SSID beaconing enabled or no administrative password set. Using a consistent configuration and employing auditing of deployed systems will help reduce this threat. Administrators should regularly audit their facilities in order to detect rogue access points and other unauthorized equipment connected to corporate networks.

Wireless IDS systems are starting to become a more common tool for protecting wireless networks. Capable of alerting administrators to attacks and rogue access points, a wireless IDS is a valuable addition to securing networks even if you are not currently deploying wireless LAN's.

Protecting Wireless Networks

- Migrate from WEP > WPA > WPA2
- Use a strong authentication mechanism such as PEAP or TTLS
- Audit network installations for consistency in deployment and configuration:
 - Identify rogue 802.11 and Bluetooth threats
 - Free and commercial tools are available
- Educate users on how to spot suspect activity on the wireless network

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Protecting Wireless Networks

A few final recommendations on securing wireless networks follow.

For 802.11 networks, use authentication methods such as PEAP or TTLS that are capable of performing mutual authentication to mitigate man-in-the-middle attacks and masquerading attacks. Migrate systems away from the legacy WEP protocol to the WPA specification immediately. Plan to migrate to the WPA2 specification to accommodate the AES-CCMP cipher as hardware replacement cycles permit.

Always require mutual authentication between clients and infrastructure equipment. Trusting that the access point, Bluetooth device, or other wireless gateway are legitimate endpoints without verifying their authenticity will likely result in compromised security for an organization.

Audit network installations to ensure that access points are deployed in a consistent manner. Eliminate the possibility of misconfigured access points with default administrator passwords, community strings, or HTTP-enabled configuration pages with consistent configurations for all equipment.

Finally, it is important to educate users on how to spot suspect activity on wireless networks. Consider the case of an attacker masquerading a hot-spot in a coffee shop. If an end user is trained to never provide authentication credentials over an unencrypted HTTP connection, he can avoid having his username and password information falling into the wrong hands.

Summary

- Wireless is popular because it unites users from the wired world
 - It might be found in multiple business units
- Popular wireless protocols include 802.11, Bluetooth, ZigBee, and WPA
- Be aware of common misconceptions in wireless security
- Follow recommended steps for planning a secure WLAN

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

Wireless networks are not going away anytime soon. Users enjoy the freedom of roaming with their laptops, handheld computers, and mobile phones. As such, wireless networks have found their way into multiple areas of businesses and corporate networks.

Popular wireless protocols include 802.11 for wireless local area networks (LANs), Bluetooth for wireless personal area networks (PANs), and WAP for mobile phone applications. Many devices even support multiple wireless protocols, such as laptops with 802.11 and Bluetooth cards or mobile phones with WAP and Bluetooth access.

Be aware of common misconceptions in wireless security. Relying on faulty information to protect the security of wireless networks can be a costly venture.

Be aware of the top four security risks for wireless networks: eavesdropping, masquerading, denial-of-service, and rogue access points.

Finally, use caution and follow recommended steps when designing wireless networks. Careful planning and execution is a critical component of protecting an enterprise from the risks associated with wireless networks.

SEC401 Installation Guide

Version Kali and Windows 8

This document covers the installation guide that helps you prepare for class. Prior to coming to class, you need to download and install the latest version of VMware Player and Kali Linux on your Windows 8 system.

Windows 8 is used as the base operating system for your laptop. Once you can run Kali Linux in VMware Player and once you can run programs in the GUI, you will be ready for class. The following steps walk you through the process.

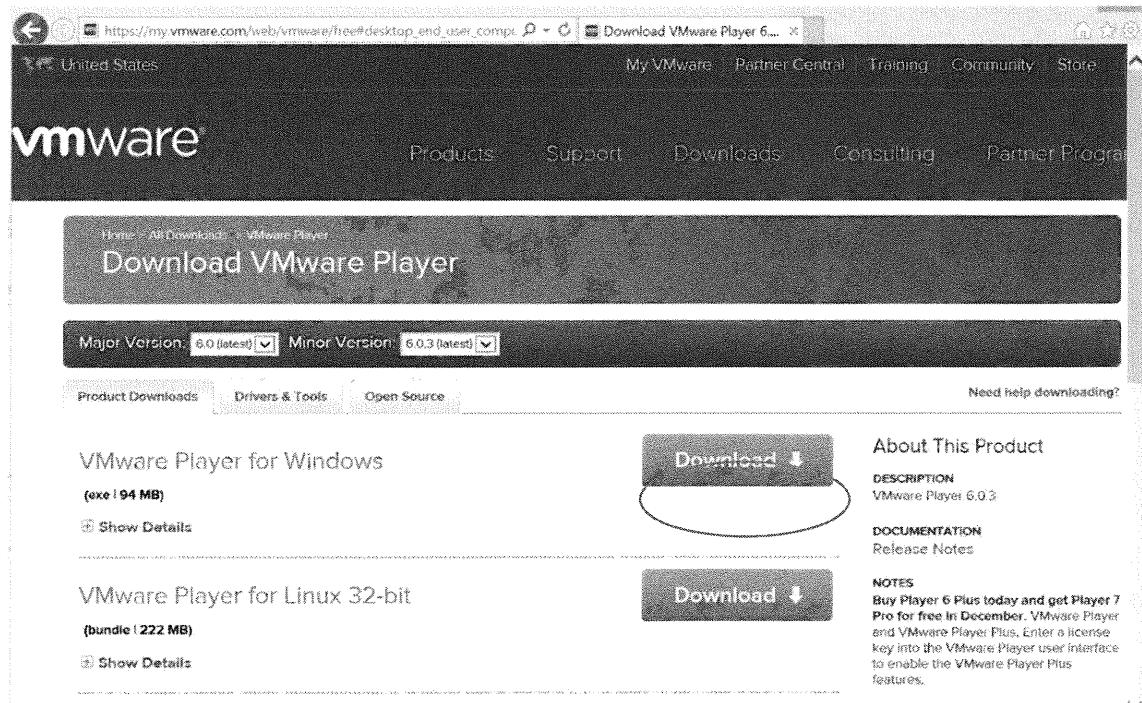
Note: If you are running a Mac OS, you can run Windows 8 on a virtual machine (VM) and Kali Linux on a separate VM. However, you need a running version of Windows 8 and a running version of Kali Linux to participate in the labs.

VMware Player

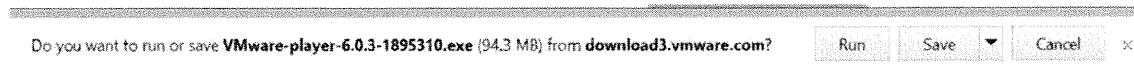
VMware Player can be downloaded and installed from www.vmware.com. The current link to download VMware Player is https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/6_0.

The links on web sites change often, so if the previous link does not work, you can find the latest version of VMware Player under Downloads, which is located at the top of the VMware web site.

On the VMware site, on the right of VMware Player for Windows, click the *Download* button.



When asked, “Do you want to run or save...?” click Run.

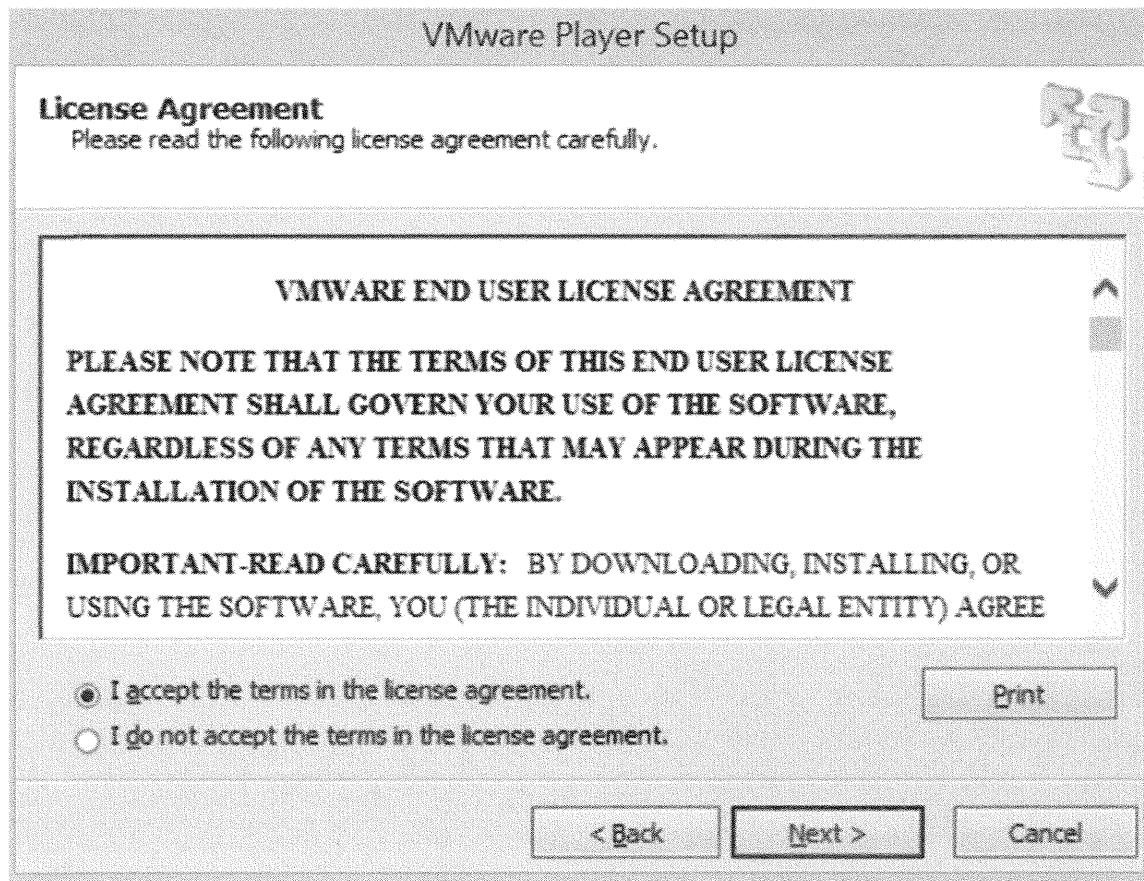


The download should begin. If you are running User Access Control (UAC) and get prompted to run the program, click Yes.

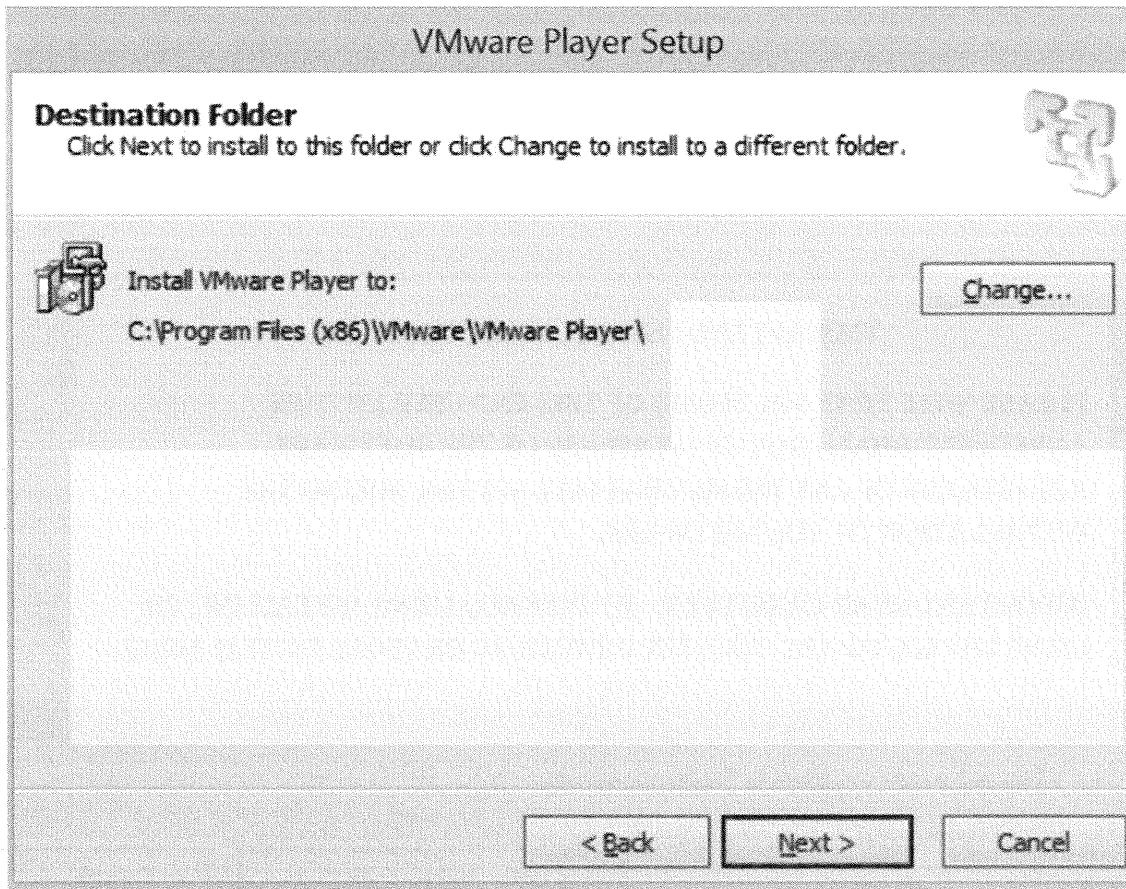
When the Welcome screen appears, click *Next* to continue.



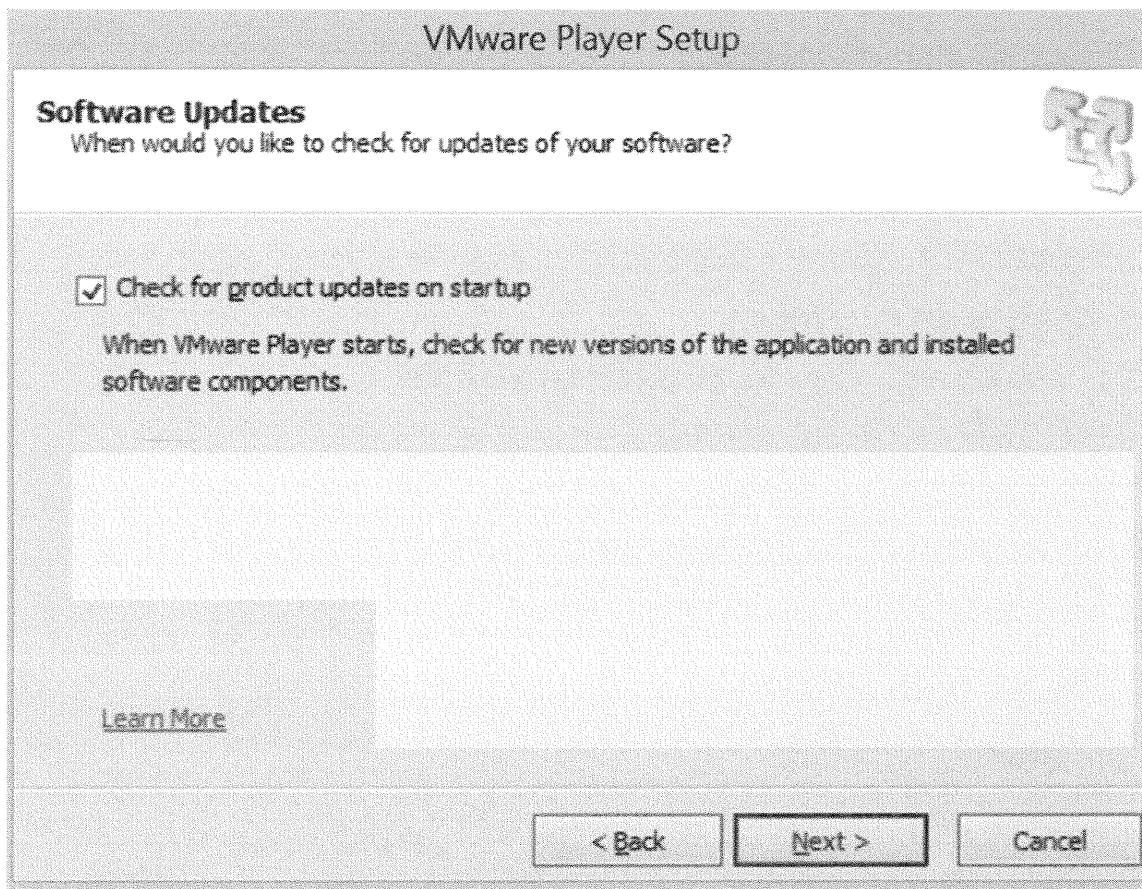
On the License Agreement page, select that *I accept the terms in the license agreement*, and then click *Next*.



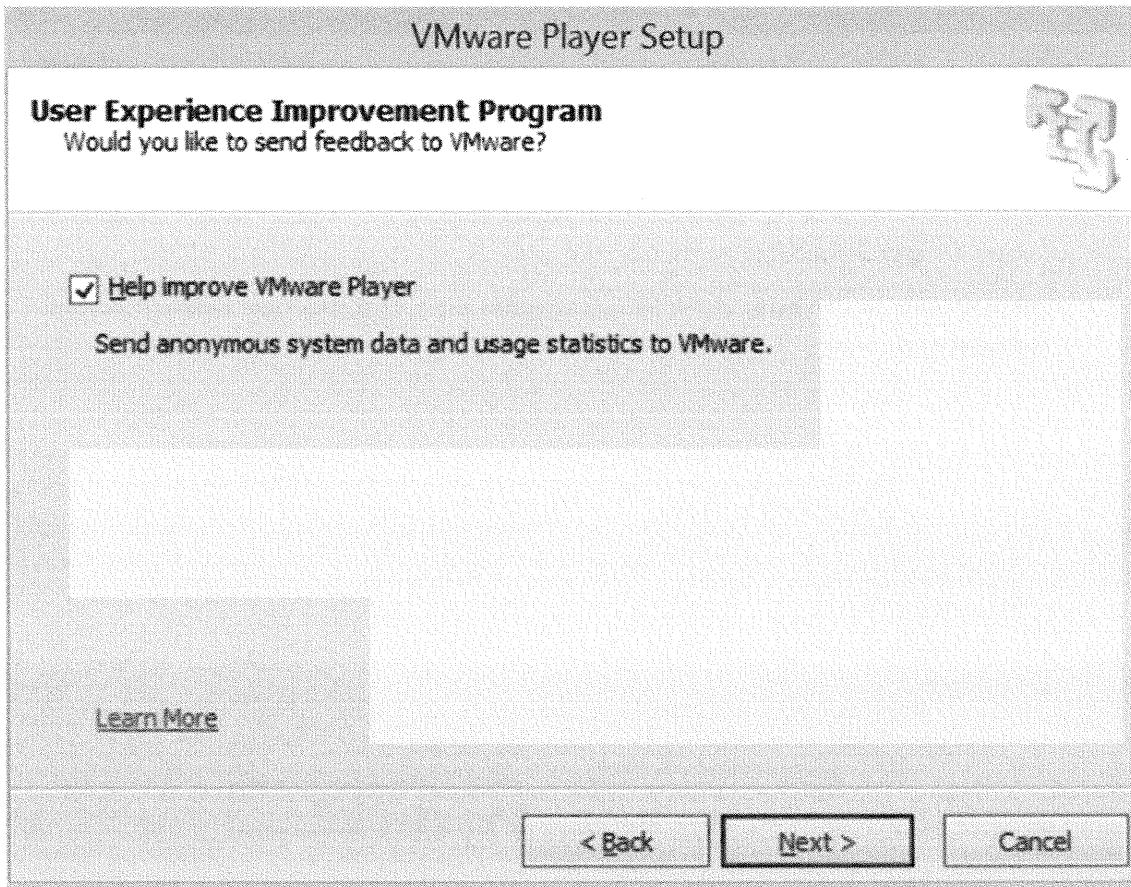
Confirm that the installation directory is correct, and then click *Next* to continue.



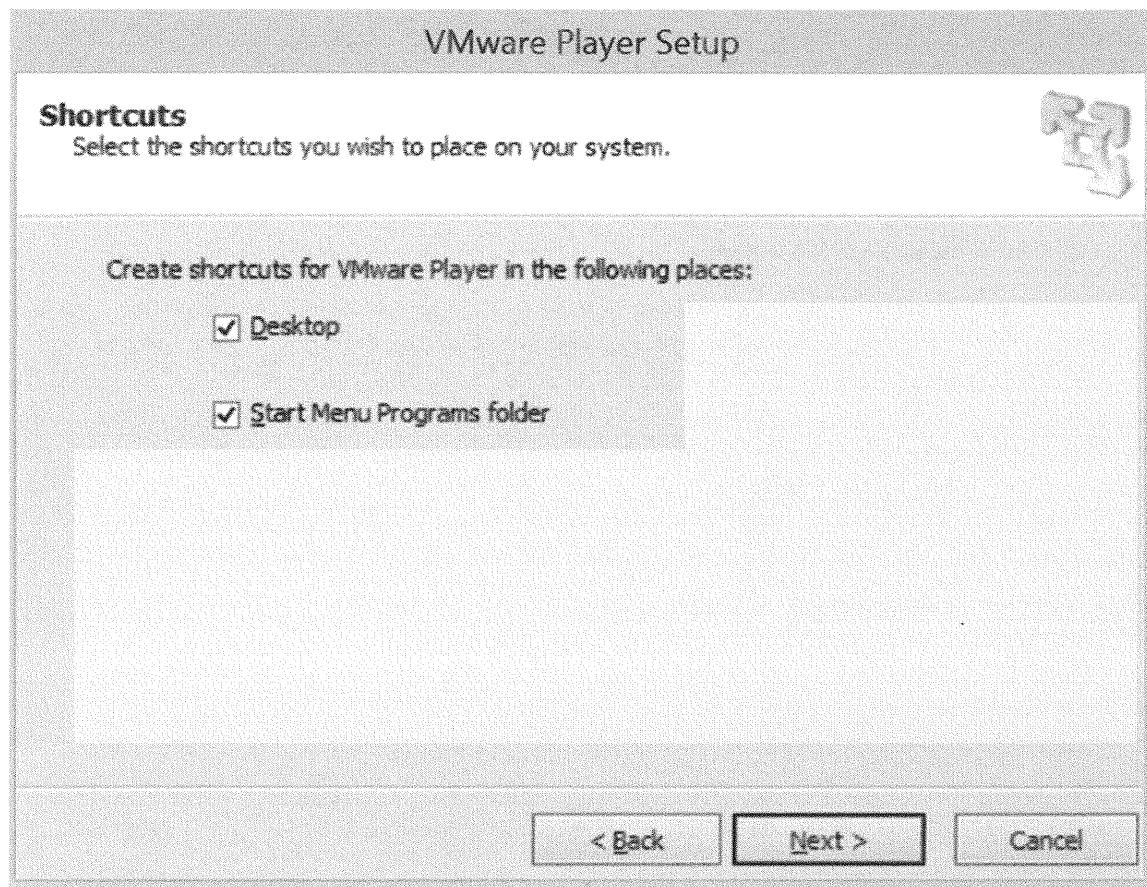
Select *Check for product updates on startup*, and then click *Next*.



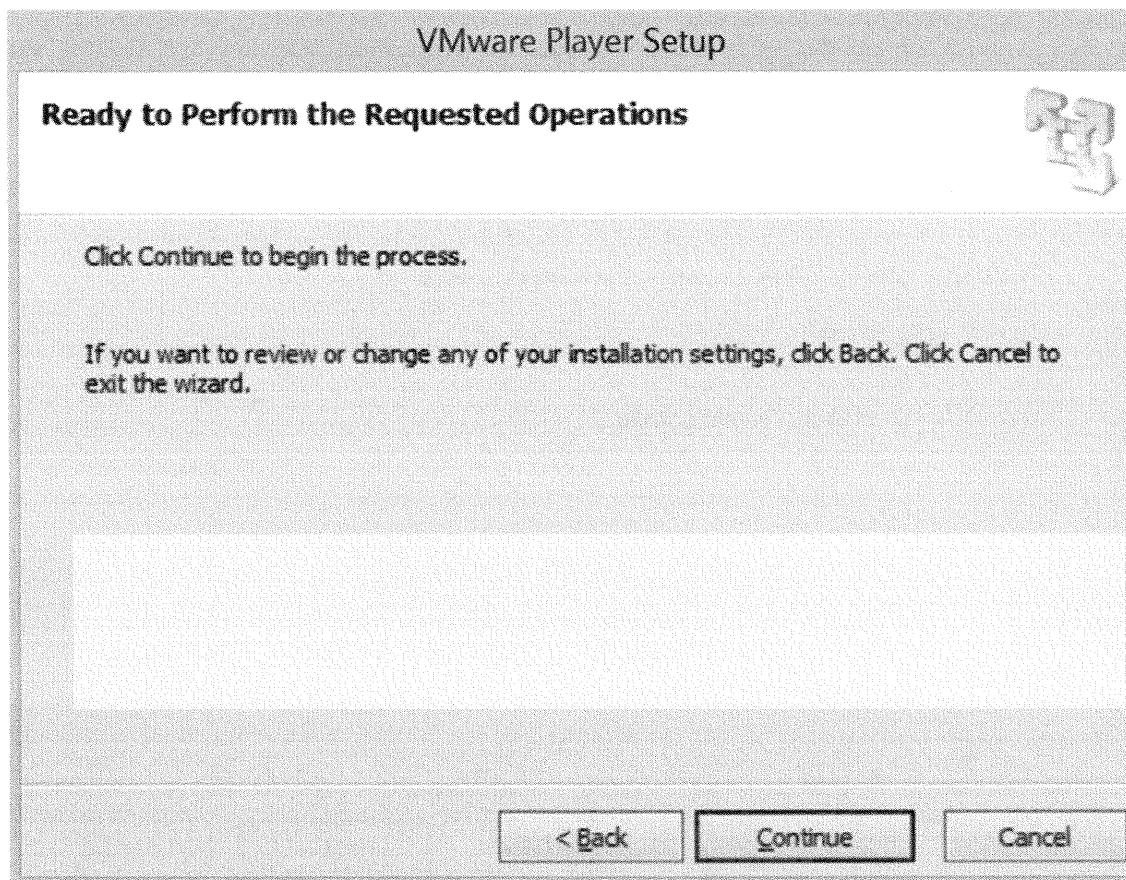
Select *Help improve VMware Player* if you want to participate in the User Experience Improvement Program, and then click *Next*.



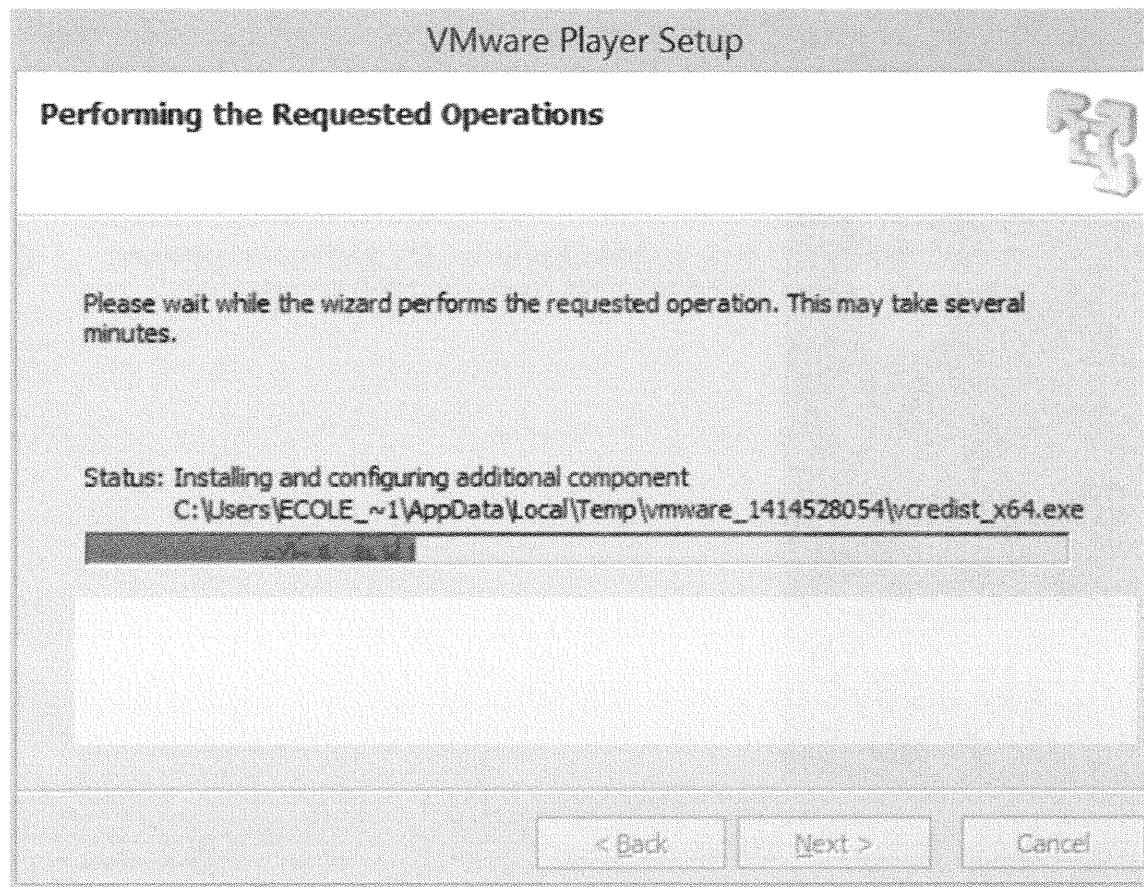
Determine which shortcuts you want to include for your system, and then click *Next*.



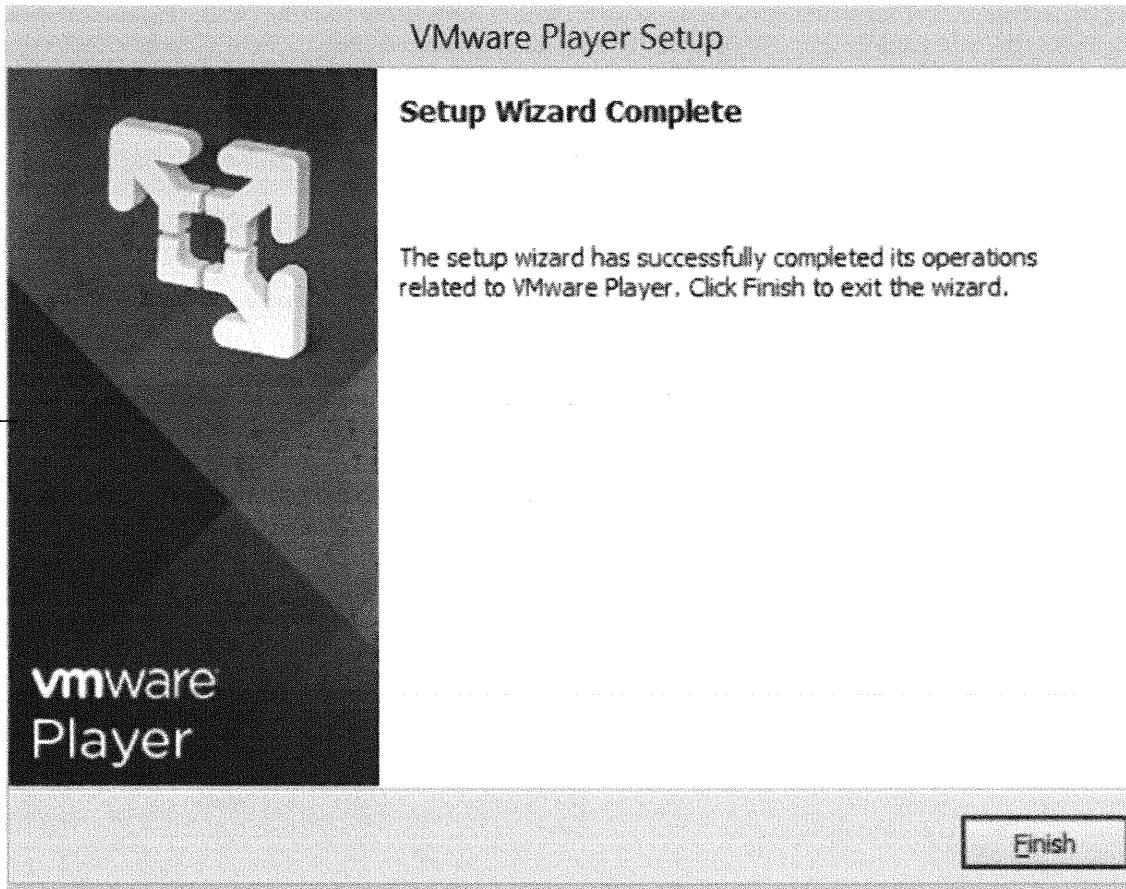
Click *Continue* to install the program.



Installation of the program begins.



Click *Finish* to complete the installation process.



VMware is now installed on your system.

Downloading Kali Linux

This section describes how to download Kali Linux on your system. Download Kali from <http://www.kali.org/>. Click *Downloads*, and then select *Custom Kali Images*.

Select either the 64-bit version or the 32-bit version, depending on the system you are going to run it on. At the left of the versions, click the + symbol for the version you want to use. This displays the options.

Kali Linux Custom VMware Images

- Kali Linux 1.0.9 VMware 64 bit (amd64)

- Download Image
- Torrent
- SHA1SUM: f1a064c41031b930ce0cdb9fd639cda5b4c6bca2

Click *Download Image*. When the Save option displays, click the *Save down arrow*, and then click *Save As* to save the file to your desktop.

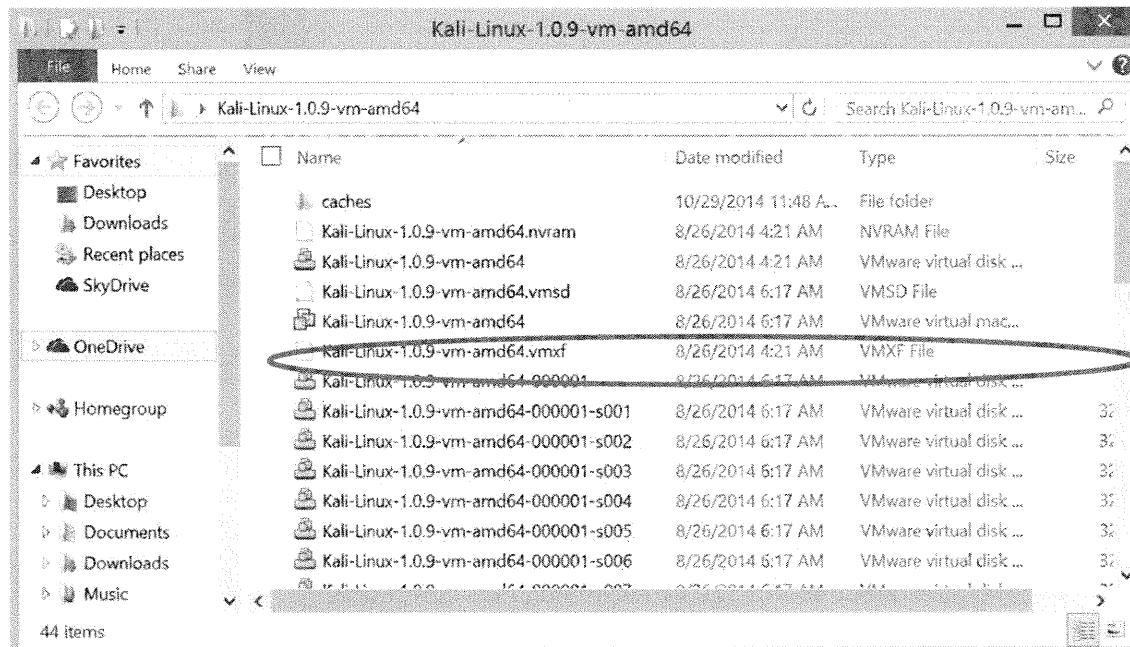


The downloaded file should now appear on the desktop. It is compressed with WinRAR. If WinRAR is not installed it can be downloaded from <http://www.rarlab.com/>. It is important to always run antivirus software and scan all files before downloading them. Web sites change and a previously trusted site could be compromised and potentially contain malware.

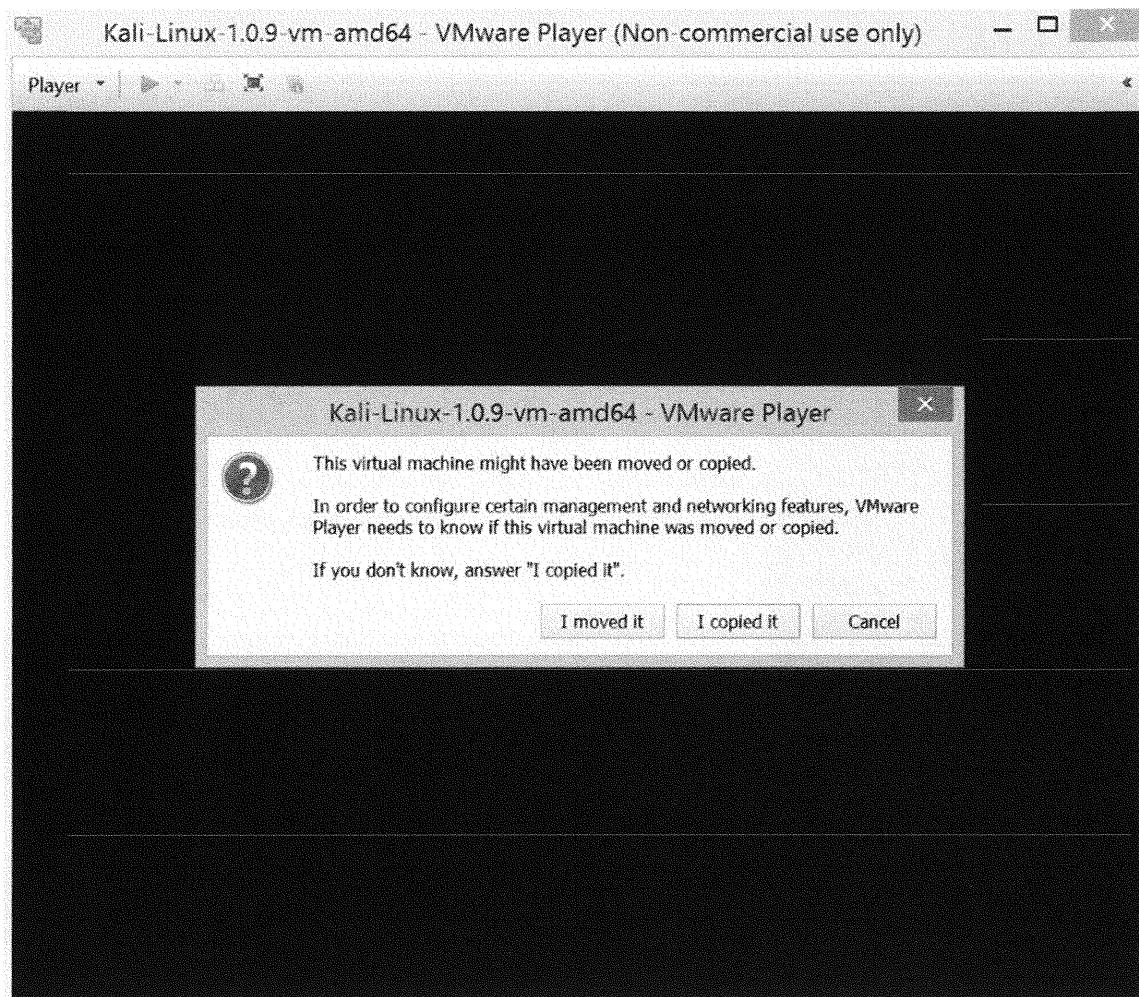
Right-click the *WinRAR* file to decompress it. Select *Extract Here* and a Kali-Linux folder will appear on the desktop. The decompression process can take 5-10 minutes based on the speed of your system. Make sure it is complete before proceeding to the next step. At this point, Kali is downloaded and ready to be run.

Running Kali

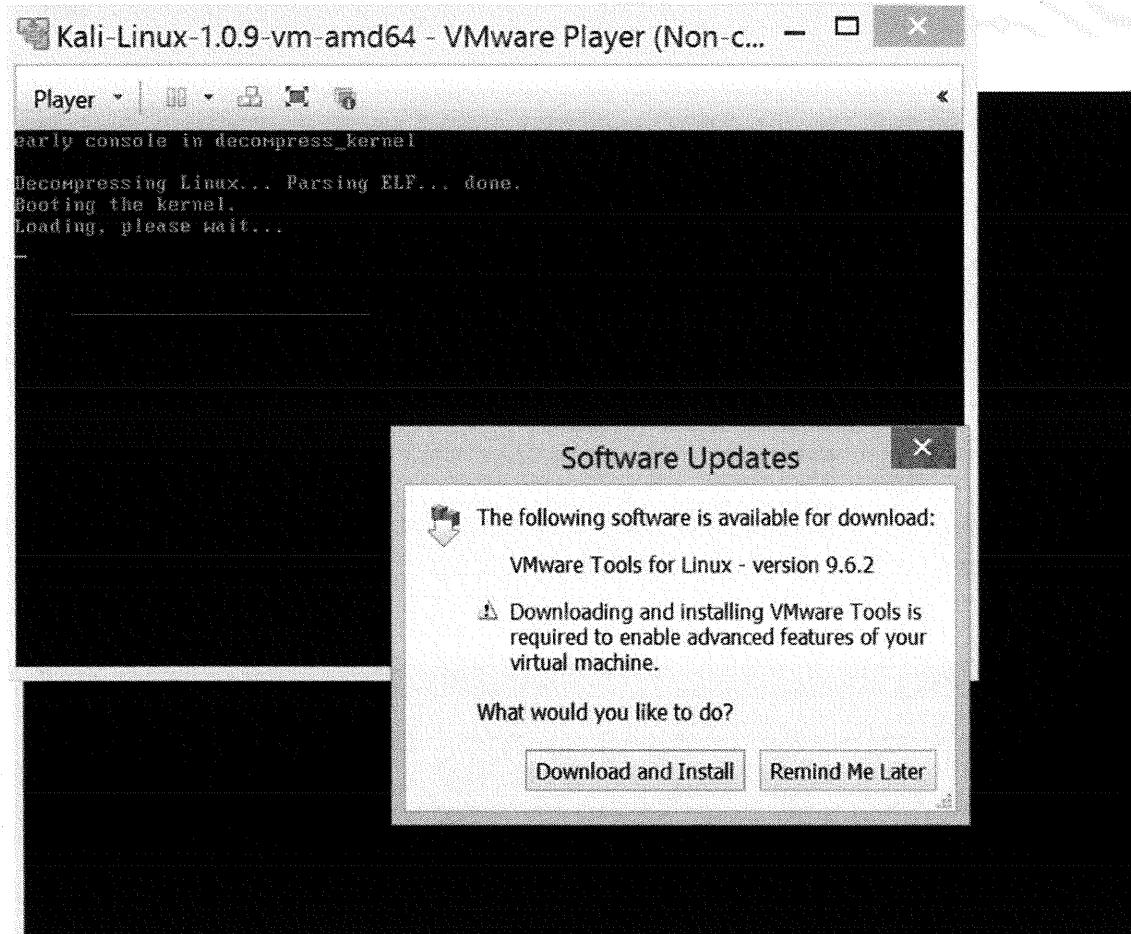
After VMware Player is installed and Kali is downloaded, open the Kali-Linux directory where the rar file was uncompressed. Double-click the file named *Kali-Linux-1.0.9-vm-amd64.vmx*; if the file extension is hidden, it can be identified by looking at the Type and by finding the VMware virtual machine file.



VMware will start. When asked whether the image was moved or copied, click *I copied it*.

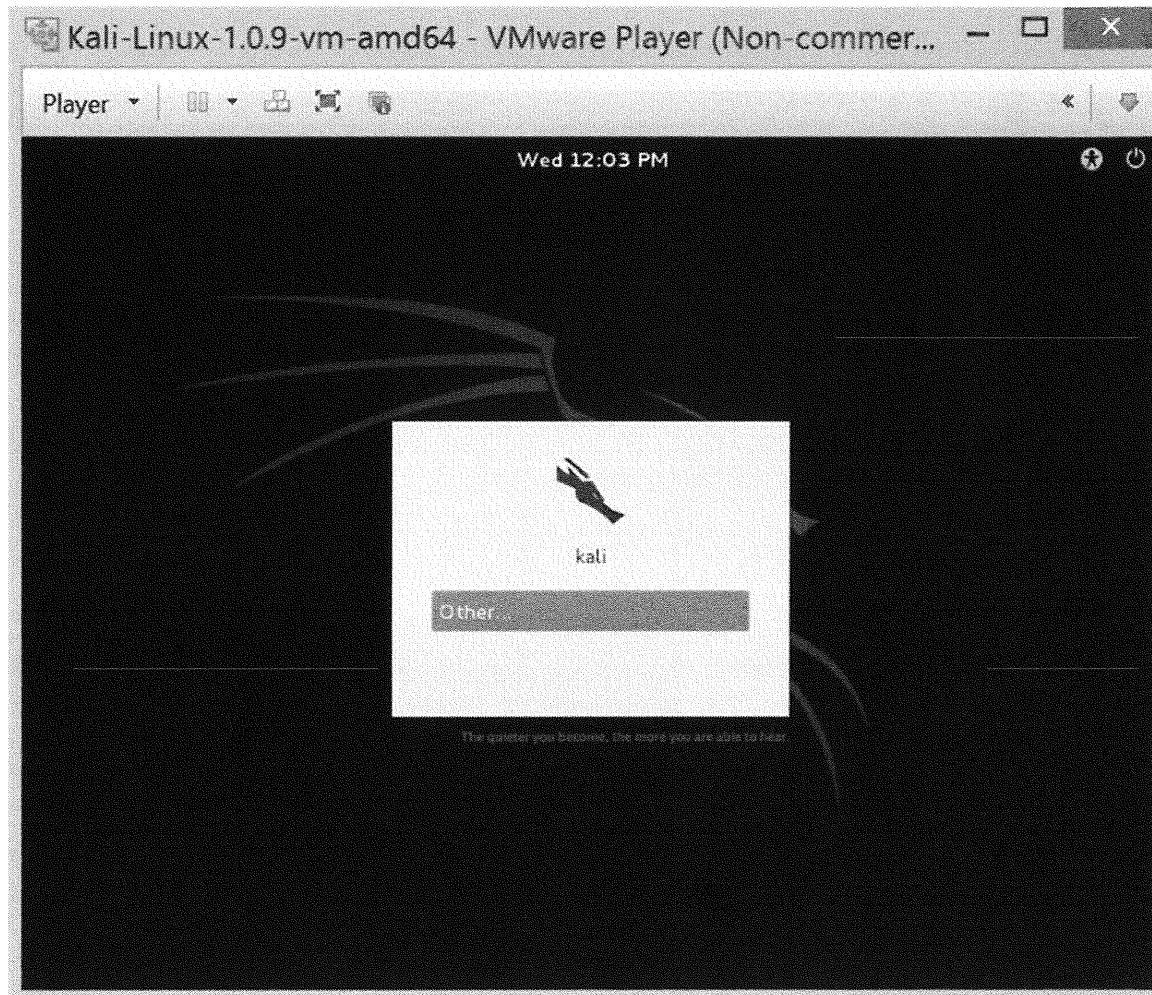


If prompted, you can download and install VMware Tools. Although the tools are not required, it is recommended you get them to optimize your use of VMware Player.

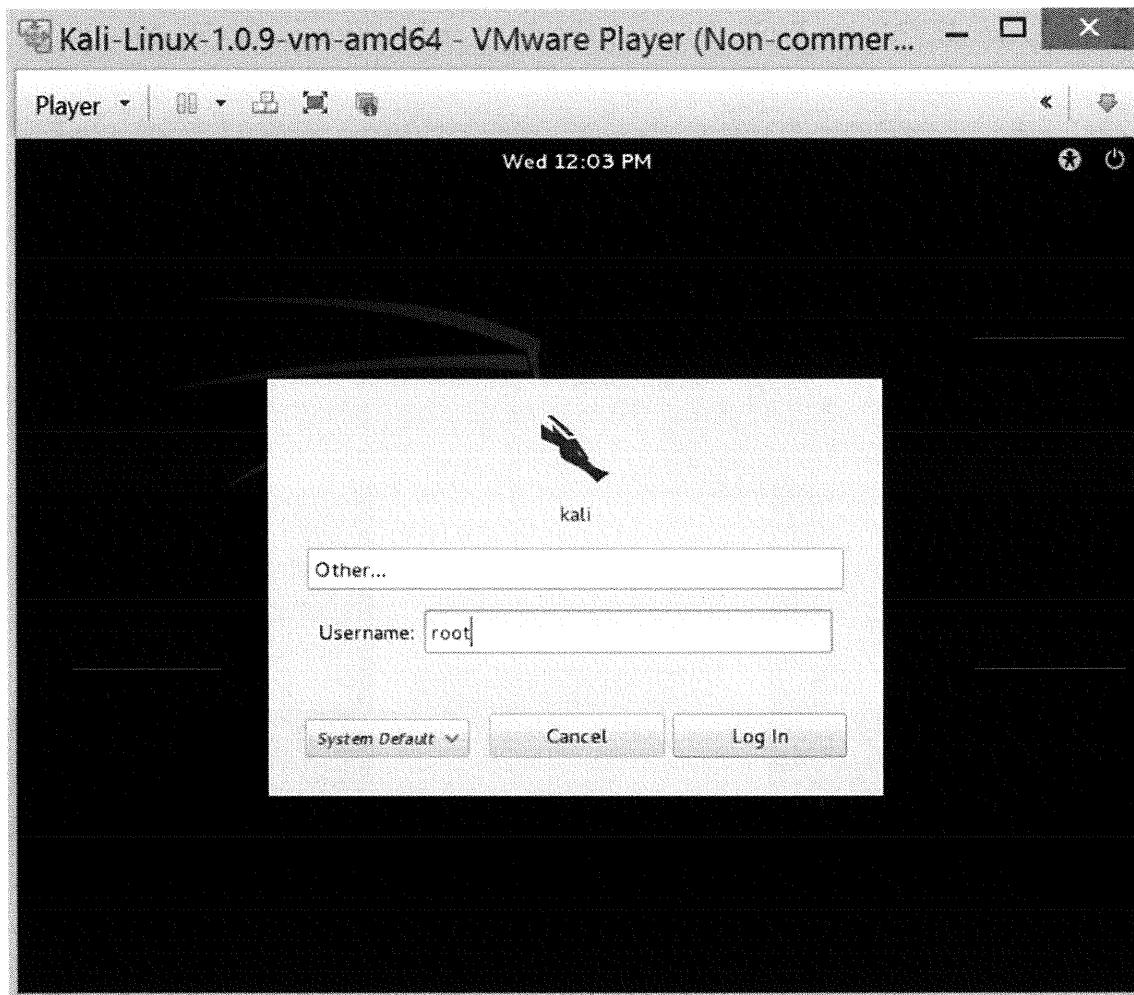


If UAC is running, click Yes to allow the program to run.

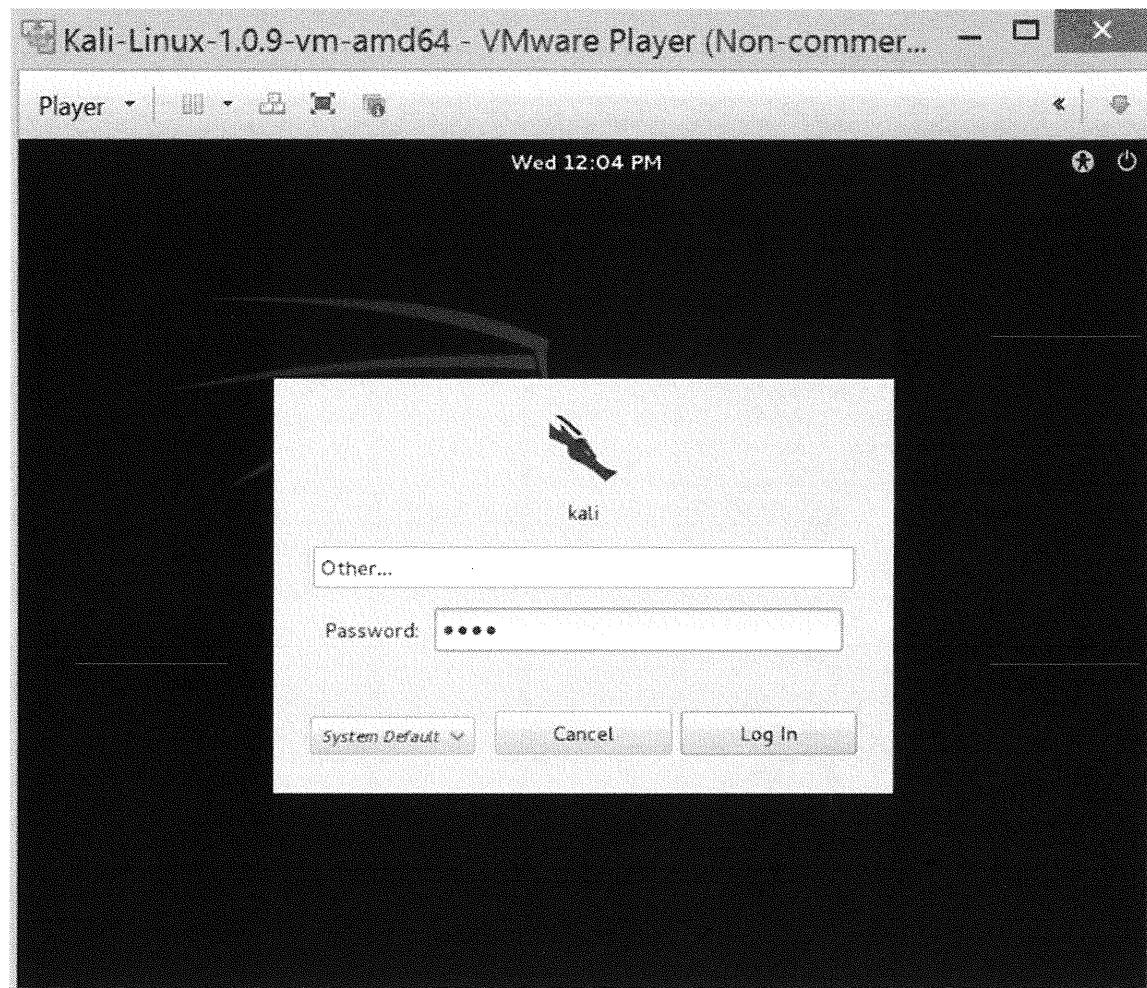
After you install VMware Tools, the system boots up and requires you to log on to it.



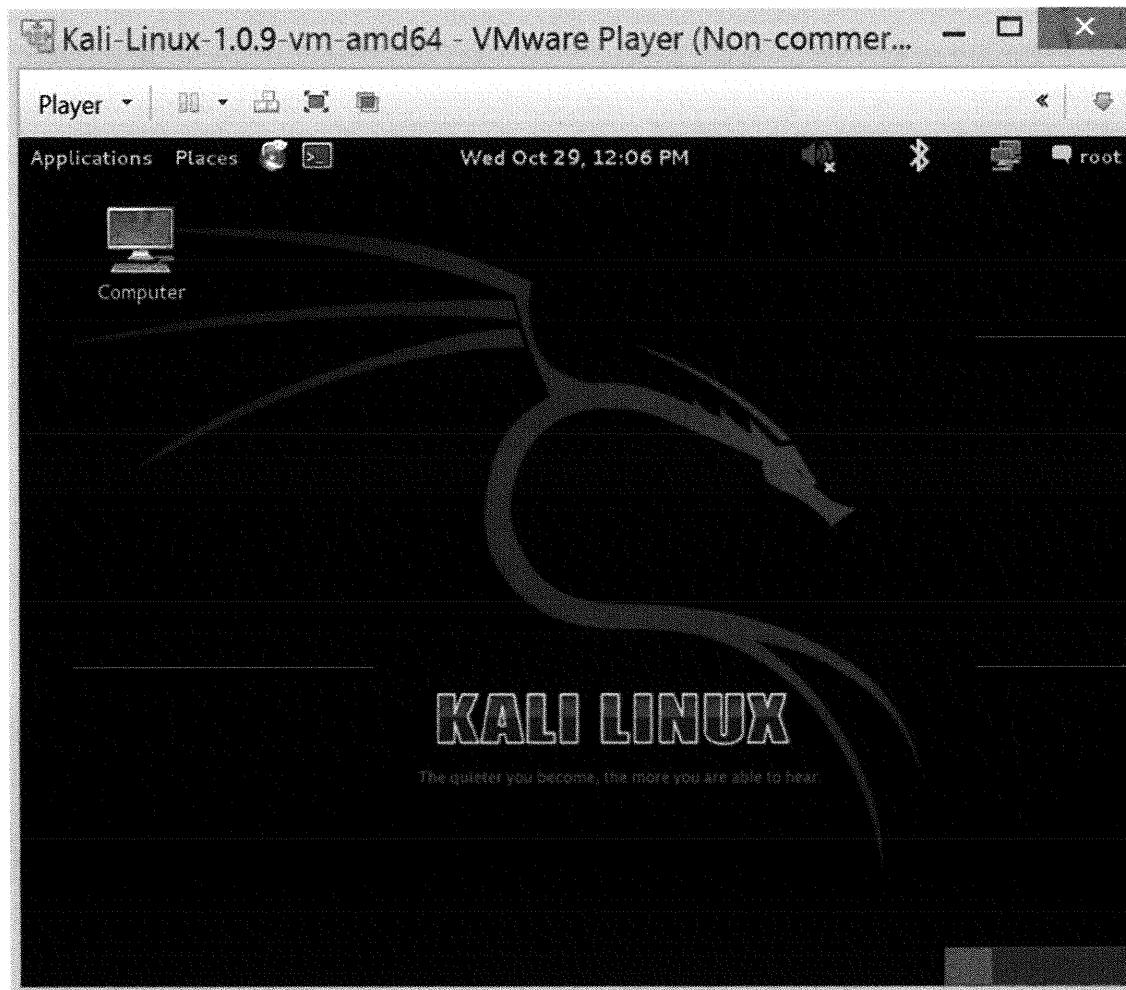
Select *Other*. Type the username **root**, and then click *Log In*.



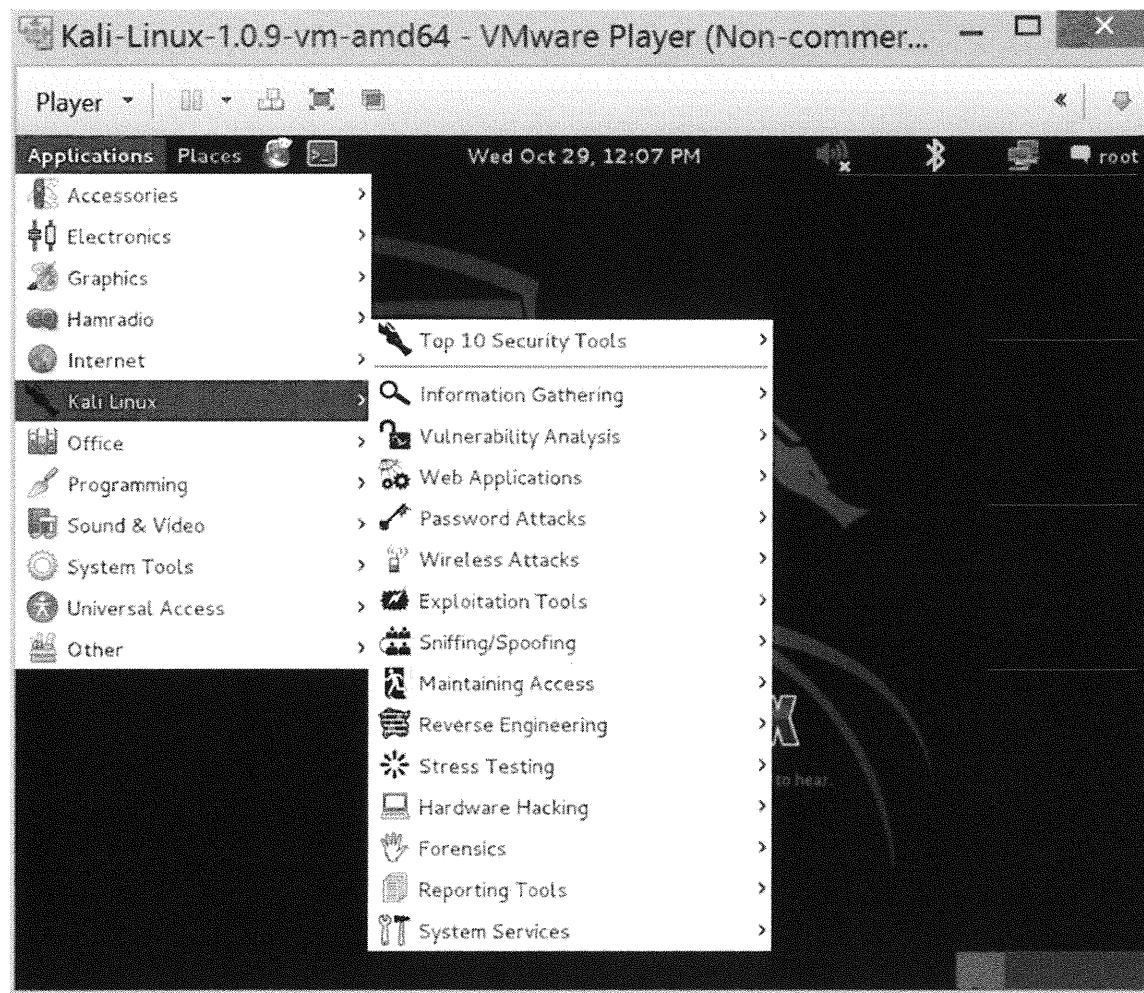
When prompted for a password, type **toor** (which is root backwards), and then click **Log In**.



Kali Linux should start, enabling you to run the labs.



At the upper, left, click *Applications* -> *Kali Linux*. You can see all of the tools that are installed.



Now you should be ready for class.

This page intentionally left blank.

SEC401 Lab Tools – Networking Concepts

SANS Security Essentials - © 2016 Secure Anchor Consulting LLC

SEC401 Lab Tools – Networking Concepts

This section intentionally left blank.

Introduction to Windows and Linux

The student will be given an overview of the Windows 8 and Kali Linux operating systems in preparation for the tools used in the SANS Security Essentials Labs.

SANS Security Essentials - © 2016 Secure Armor Consulting LLC

Introduction to Operating Systems

It is important that you become familiar with Windows and Linux in preparation for this course. The exercises in this book assume that you have a basic knowledge of both of these operating systems. Although this chapter provides an overview of both operating systems, it is not intended as a comprehensive guide to Windows and Linux; it is intended to help prepare you for this course.

Windows (1)

- Understand the cmd prompt and critical commands including:
 - cmd
 - ipconfig
 - regedit
 - netstat
 - cls
 - dir
 - mkdir
 - Task Manager

SANS Security Essentials - © 2010 Secure Anchor Consulting, LLC

Windows (1)

The Windows operating system is a dynamic and continually changing operating system with new security patches and hot fixes being released often. In a typical production environment, you should maintain a patching schedule to keep your systems up-to-date.

This "Introduction to Windows and Linux" guide gives you the basic commands and actions you need to know for the Security Essentials Labs. This document introduces you to several commands: cmd, ipconfig, regedit, netstat, cls, dir, and mkdir. It also introduces you to the Task Manager.

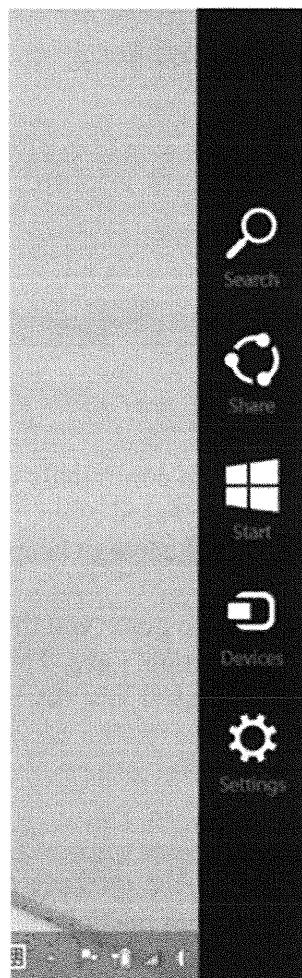
The Cmd Prompt

With the release of Windows 2000 Professional, the cmd.exe program replaced the previous 16-bit command.com program. There are many benefits of using cmd, including the following:

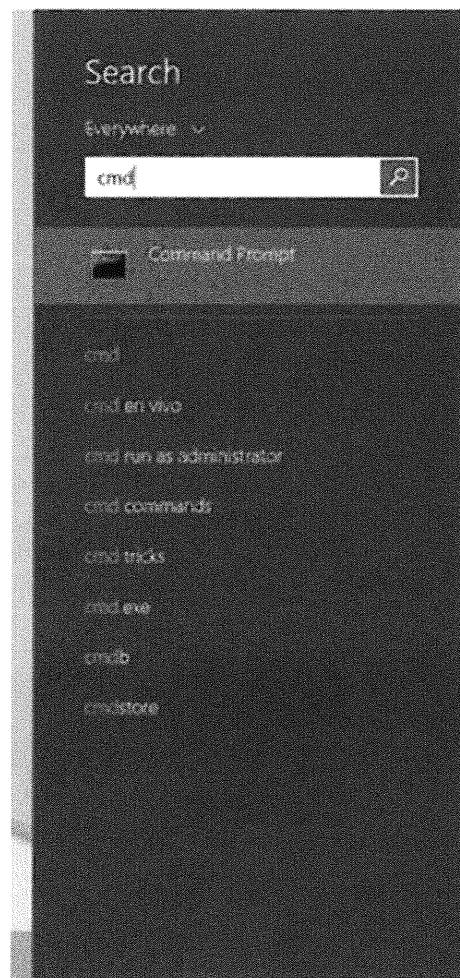
- The capability to run scripts in the CMD language
- The Windows Management Instrumentation Console (WMIC)
- No 8.3 filename limitations
- The capability to run multiple commands on the same command line
- Support for command pipelines
- Help functionality with /?

The following list of tasks shows you how to use the command prompt to obtain help or information about your system:

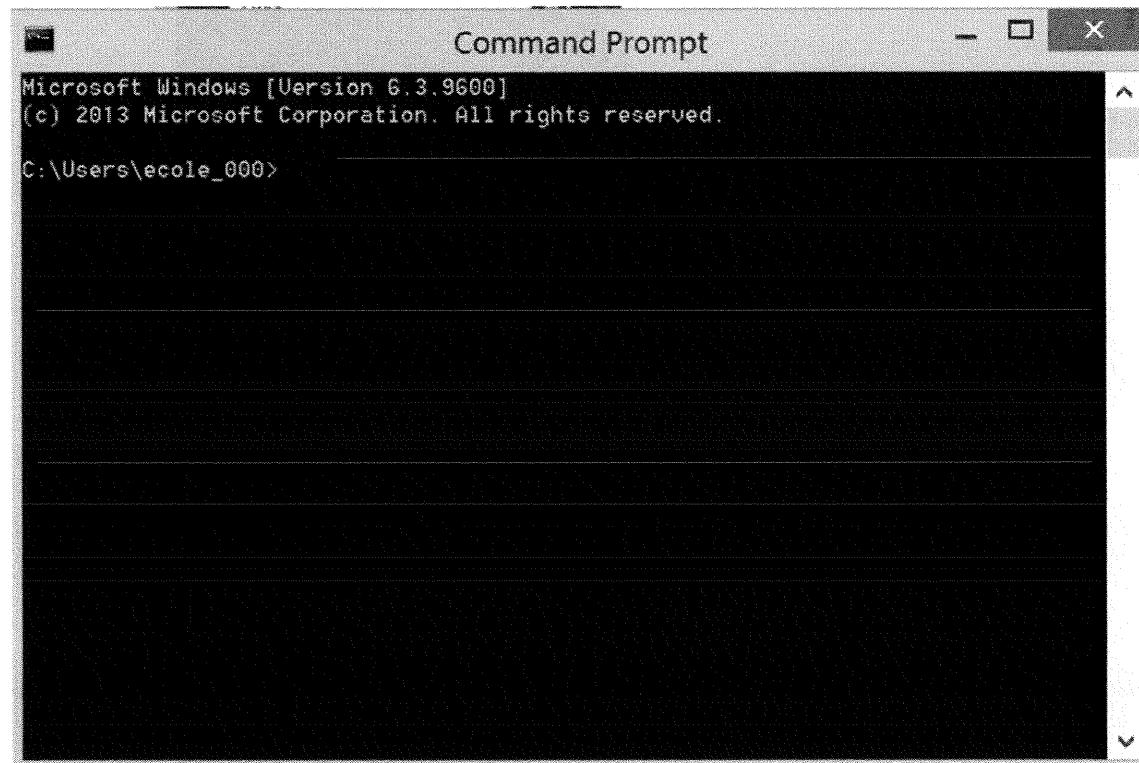
1. To display the command prompt, in Windows 8, from the right side of your screen, swipe in to display the side menu.



2. Click *Search*, type **cmd**, and then click the *Command Prompt* icon.



3. The command prompt opens.



4. If you need help with a command while using cmd, after the command in question, type the following:

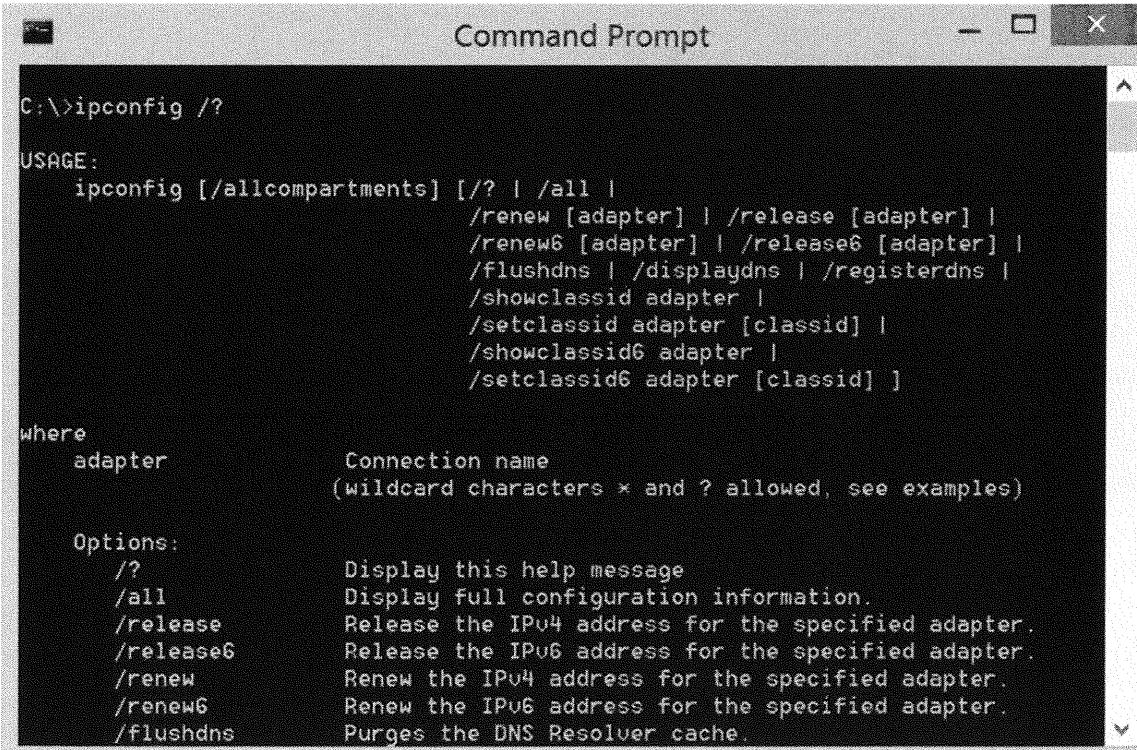
/?

For example, to obtain NIC TCP/IP information, type the following:

Ipconfig

For a list of the available ipconfig options, after the command prompt, type the following:

ipconfig /?



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command entered is "C:\>ipconfig /?". The output provides usage information for the ipconfig command, including options like /all, /renew, and /release, and details about connection names and adapter options.

```
C:\>ipconfig /?

USAGE:
  ipconfig [/allcompartments] [/? | /all | /renew [adapter] | /release [adapter] | /release6 [adapter] | /flushdns | /displaydns | /registerdns | /showclassid adapter | /setclassid adapter [classid] | /showclassid6 adapter | /setclassid6 adapter [classid] ]

where
  adapter           Connection name
                    (wildcard characters * and ? allowed, see examples)

Options:
  /?                Display this help message
  /all              Display full configuration information.
  /release          Release the IPv4 address for the specified adapter.
  /release6         Release the IPv6 address for the specified adapter.
  /renew            Renew the IPv4 address for the specified adapter.
  /renew6           Renew the IPv6 address for the specified adapter.
  /flushdns         Purges the DNS Resolver cache.
```

5. To identify the IP address information for your system, type the following:

ipconfig /all

This command also displays your MAC address, as shown in the following screen. If you have a virtual machine application installed, you will receive other interfaces in addition to the following.

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window displays the output of the "ipconfig /all" command. The output includes system configuration details and network adapter configurations for three adapters: Local Area Connection 3 (Wireless LAN), Microsoft Wi-Fi Direct Virtual Adapter, and Bluetooth Network Connection. All three adapters show a "Media State" of "Media disconnected".

```
C:\>ipconfig /all
Windows IP Configuration

Host Name . . . . . : EricSurface1
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled . . . . . : No
WINS Proxy Enabled . . . . . : No
DNS Suffix Search List . . . . . : hsd1.va.comcast.net

Wireless LAN adapter Local Area Connection 3:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :
    Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
    Physical Address . . . . . : 2A-18-78-CB-72-CC
    DHCP Enabled . . . . . : Yes
    Autoconfiguration Enabled . . . . . : Yes

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :
```

Windows (2)

- Understand the registry and how to edit it using regedit
- Learn how to change IP addresses through network properties
- Learn how to connect to shares
- Use Task Manager
- Set up directories

SANS Security Essentials - © 2010 Secure Anchor Consulting, LLC

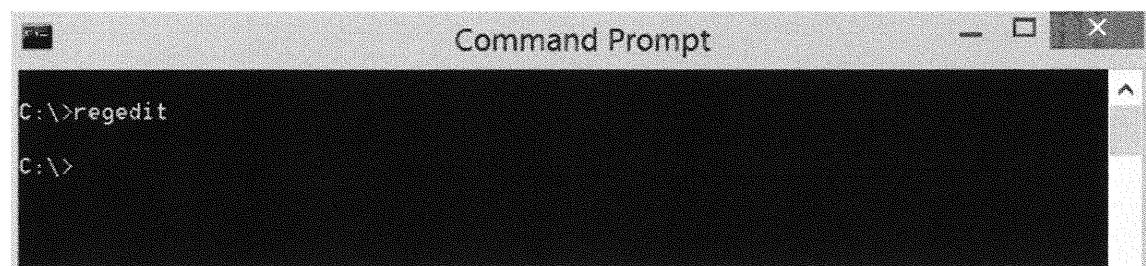
Windows (2)

To edit the Registry in a Windows environment, at the Run prompt, you can use the regedit command. An advantage of using regedit is that you can search every hive for specific keys, values, and data.

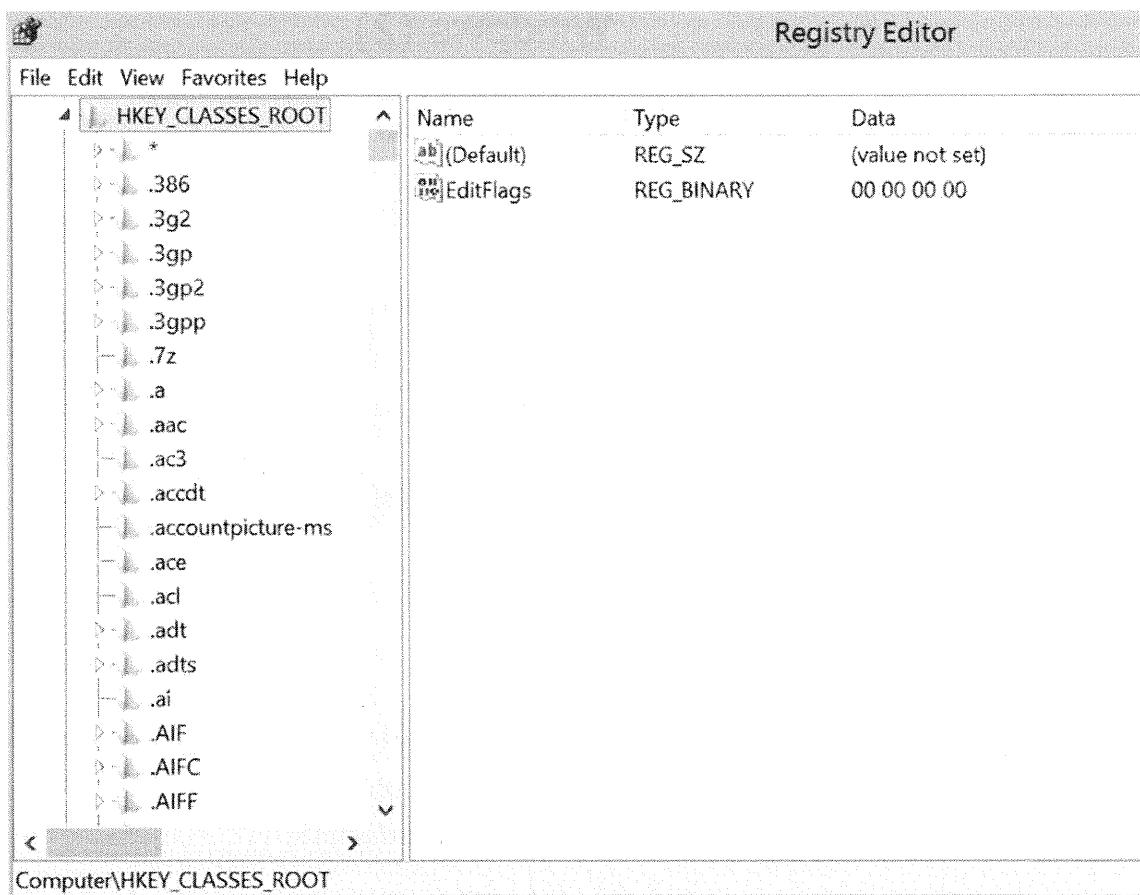
Warning: When using regedit, exercise extreme caution because any change you make is permanent and can potentially render your system unusable.

To edit the Registry and use regedit, perform the following tasks:

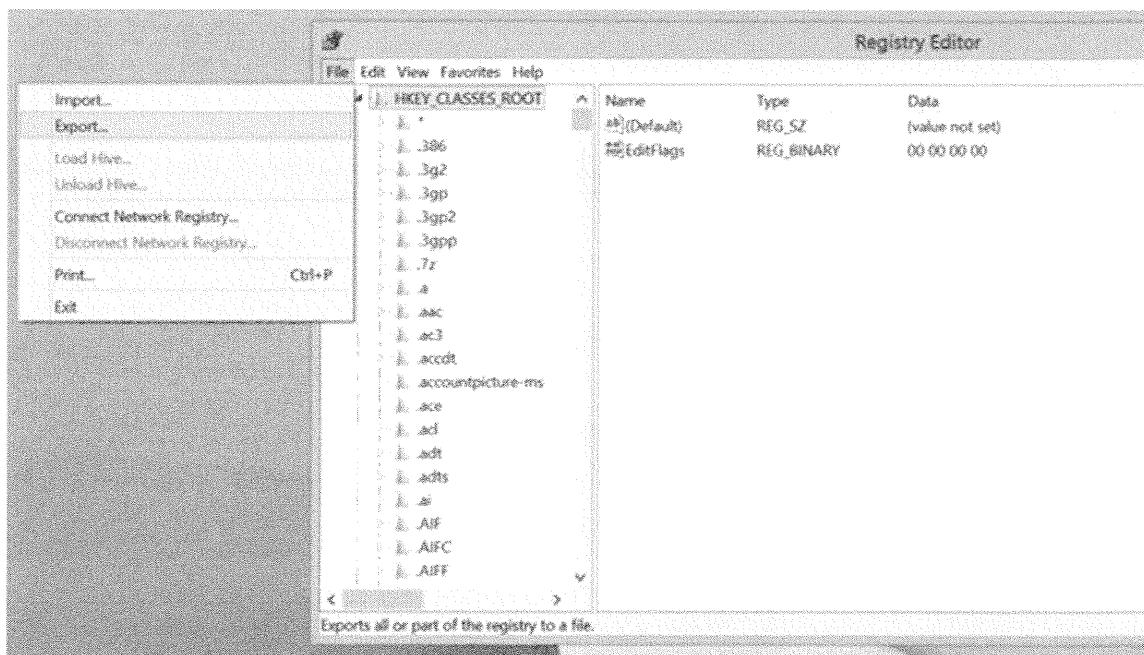
1. To start regedit, from the command prompt, type **regedit** and press *Enter*.



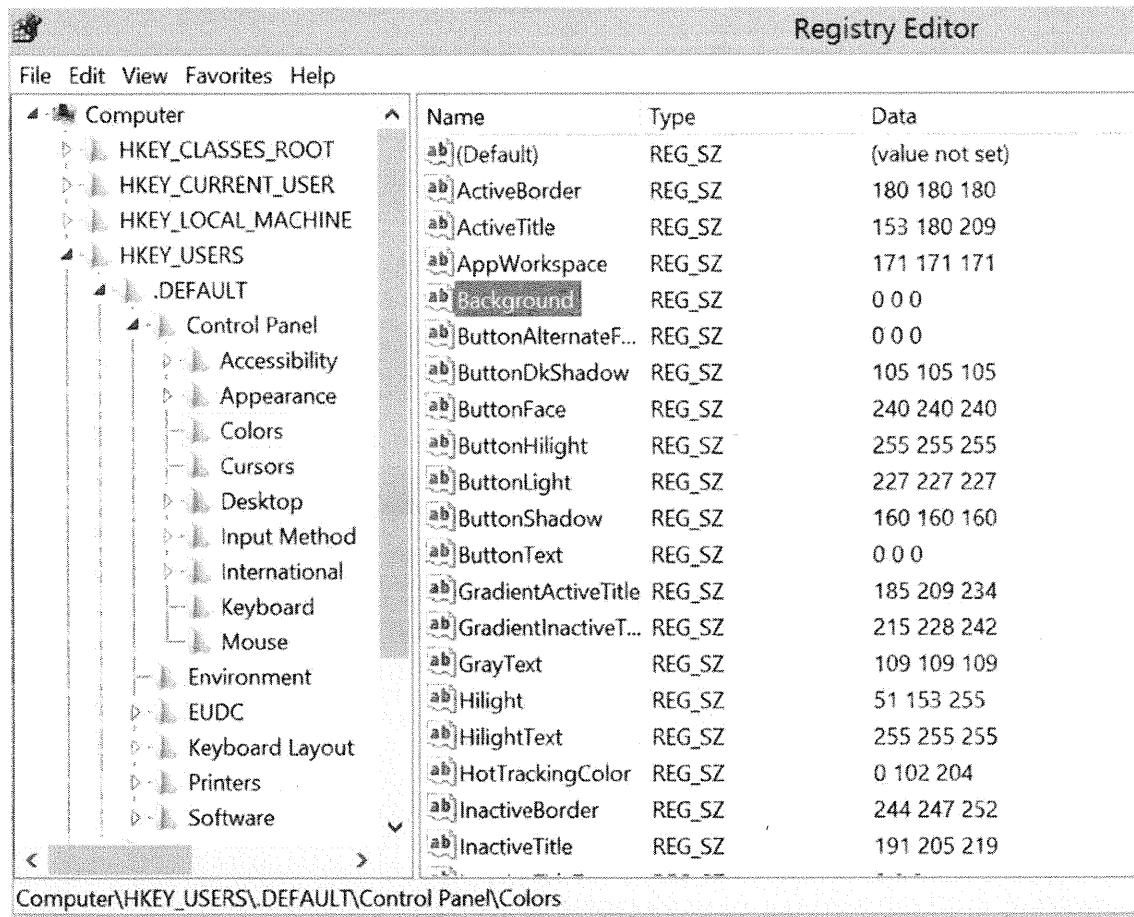
2. Regedit starts, as shown in the following.



3. When editing the Registry, you should always save a copy of the keys you change so that you can undo mistakes. To do this, from the File menu, click *Export*. Select a location to save the Registry to and then *Save* the Registry.



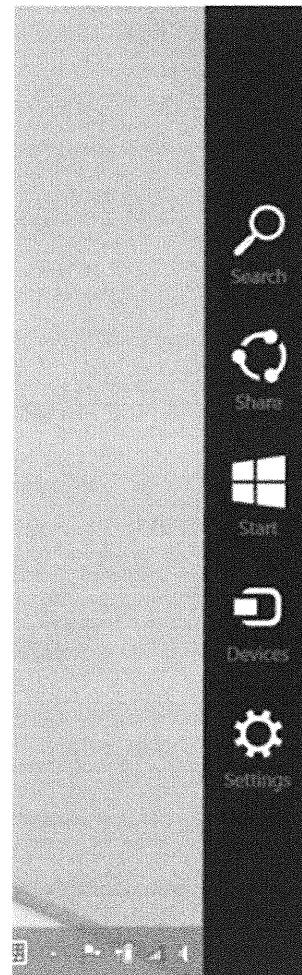
4. There are a lot of powerful things you can do with the Registry. To minimize any impact on your system, let's look at a relatively safe example. To change the logon screen background color, you can edit **HKEY_USERS -> DEFAULT -> Control Panel -> Colors**. You can change the color using RGB settings. For example, 0 0 0 is black and 255 255 255 is white. If you change the color, you need to reboot the system for the change to apply. This is a basic example, but it is not recommended that you change the color at this time.



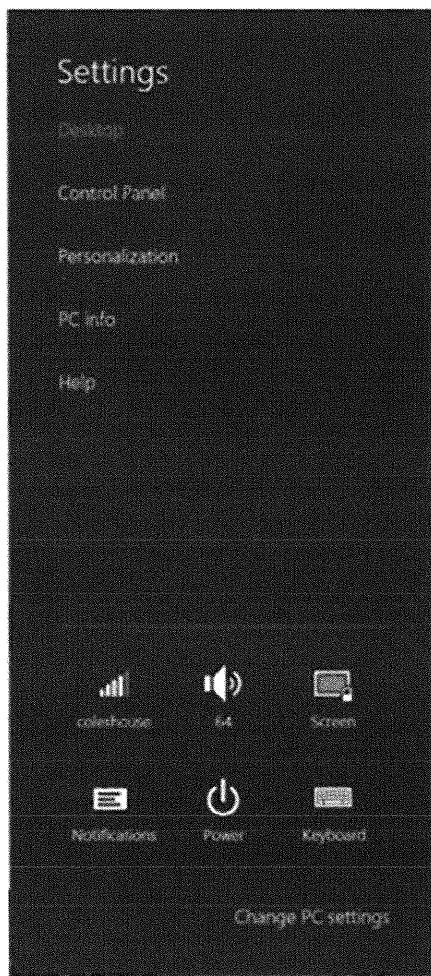
IP Changes

To make IP changes on your Windows system, perform the following steps:

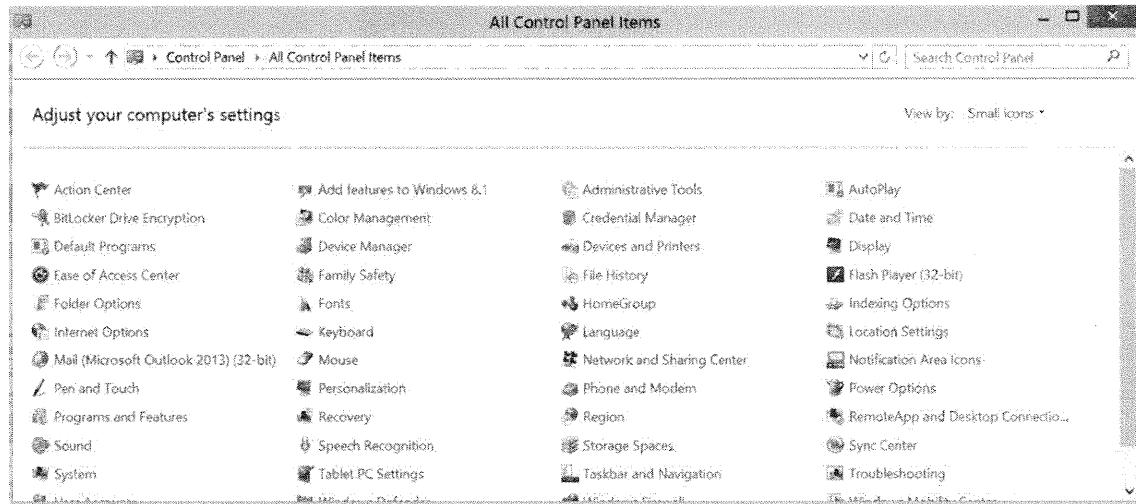
1. To make IP address changes to your local machine, open the NIC properties. To open the menu, right-swipe the screen.



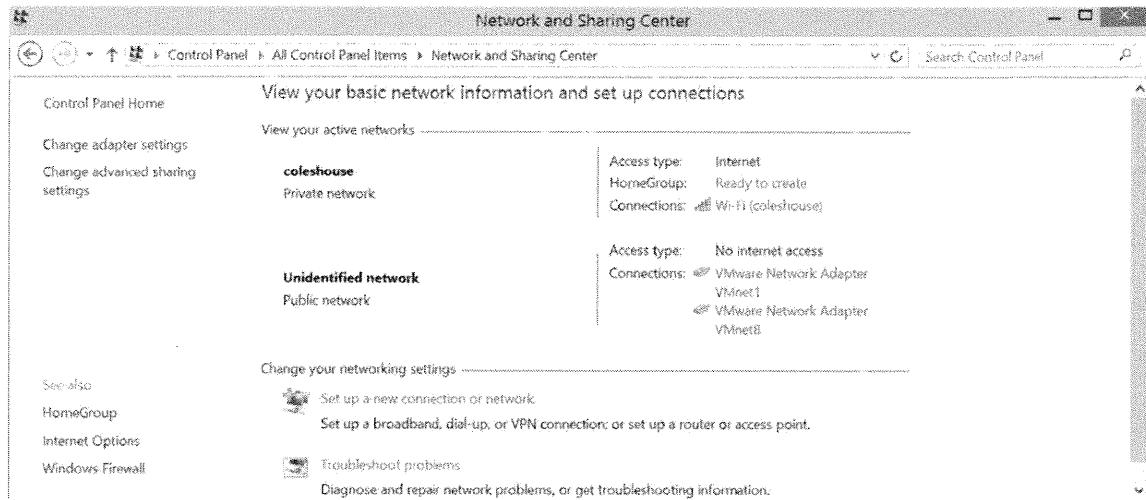
2. Click *Settings*.



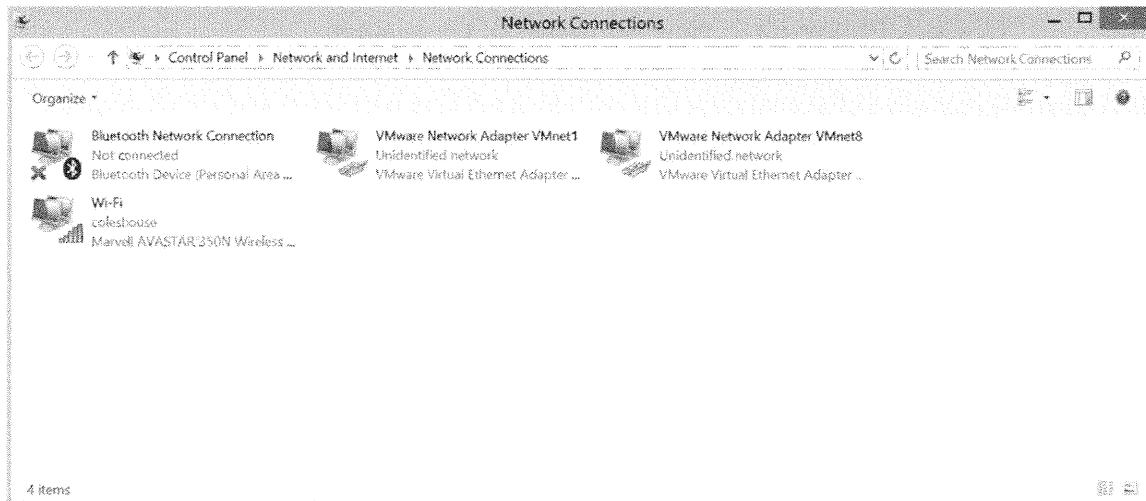
3. To open Control Panel, from Settings, click *Control Panel*.



4. Click *Network and Sharing Center Internet*.



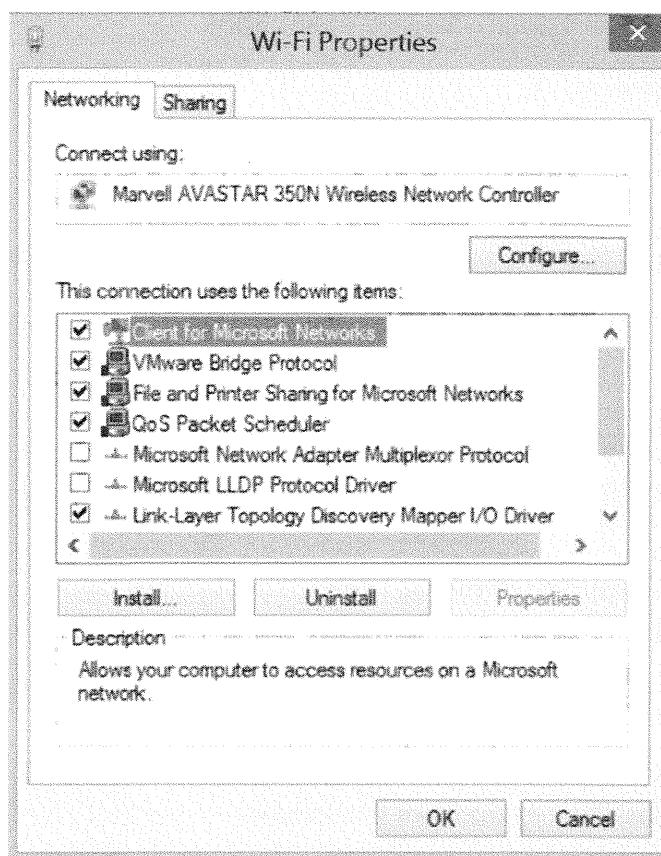
5. Click *Change adapter settings* to view the current network adapters on your system.



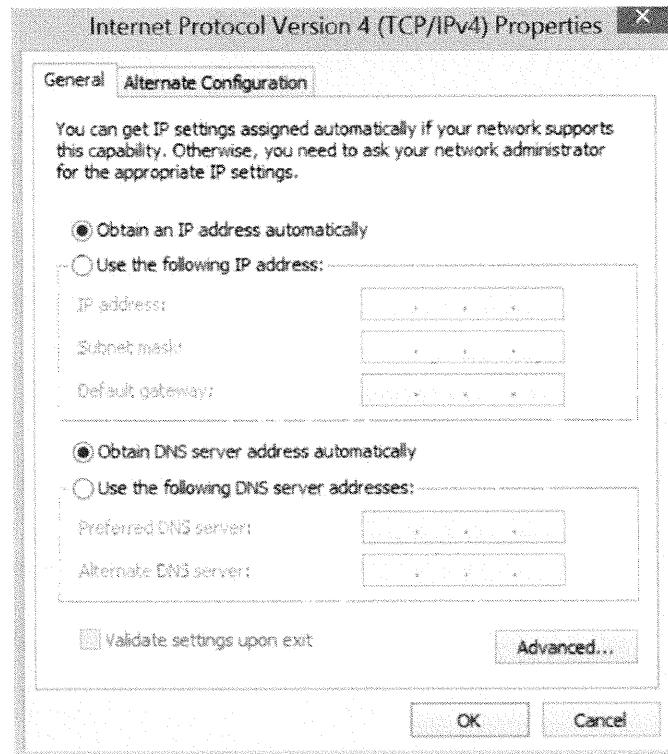
6. Double-click an adapter to view/change the settings. For example, double-click the *Wi-Fi adapter* to view details about the adapter.



7. Click *Properties* to view the details of the adapter.



8. Highlight *Internet Protocol (TCP/IP)* and click the *Properties* button. Most systems use DHCP, but this is where you can change an address. This can also be used as a basic way to spoof an IP address for certain types of attacks.



9. Click *Cancel* when you are done to close out the window.

Viewing Ports

To see what ports are open on your box, you can use the netstat command from a command prompt, as shown in the following screen. Because you are not connected to a network, you might receive limited information. To run this command, open a command prompt and type the following:

```
netstat -an
```

Command Prompt

```
C:\>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135           0.0.0.0:0             LISTENING
  TCP    0.0.0.0:445           0.0.0.0:0             LISTENING
  TCP    0.0.0.0:902           0.0.0.0:0             LISTENING
  TCP    0.0.0.0:912           0.0.0.0:0             LISTENING
  TCP    0.0.0.0:5357          0.0.0.0:0             LISTENING
  TCP    0.0.0.0:49152          0.0.0.0:0             LISTENING
  TCP    0.0.0.0:49153          0.0.0.0:0             LISTENING
  TCP    0.0.0.0:49154          0.0.0.0:0             LISTENING
  TCP    0.0.0.0:49155          0.0.0.0:0             LISTENING
  TCP    0.0.0.0:49156          0.0.0.0:0             LISTENING
  TCP    0.0.0.0:49211          0.0.0.0:0             LISTENING
  TCP    127.0.0.1:51613         127.0.0.1:51614        ESTABLISHED
  TCP    127.0.0.1:51614         127.0.0.1:51613        ESTABLISHED
  TCP    127.0.0.1:51625         127.0.0.1:51626        ESTABLISHED
  TCP    127.0.0.1:51626         127.0.0.1:51625        ESTABLISHED
  TCP    192.168.1.129:139        0.0.0.0:0             LISTENING
  TCP    192.168.1.129:51617       104.64.64.23:443      CLOSE_WAIT
  TCP    192.168.1.129:51620       208.91.0.10:443       CLOSE_WAIT
  TCP    192.168.1.129:51633       23.13.68.236:80       CLOSE_WAIT
```

You can see every open port and the state (listening, waiting, or connected) of each port. The netstat command also shows you TCP and UDP connections.

Although netstat shows you which ports are open, by default, it does not show you which service is causing a given port to be open. Attackers connect to systems via ports. The more ports that are open, the more avenues of attack. Therefore, it is important to shut down unneeded ports. To close a port, you need to know which service is causing a given port to be open. Typing the command **netstat -o** shows which service is causing a given port to be open. If you do not have an active network connection, you might receive limited information.

```
Command Prompt
C:\>netstat -o

Active Connections

  Proto  Local Address          Foreign Address        State      PID
  TCP    127.0.0.1:51613        EricSurface1:51614    ESTABLISHED 5316
  TCP    127.0.0.1:51614        EricSurface1:51613    ESTABLISHED 5316
  TCP    127.0.0.1:51625        EricSurface1:51626    ESTABLISHED 5264
  TCP    127.0.0.1:51626        EricSurface1:51625    ESTABLISHED 5264
  TCP    192.168.1.129:51617    a104-64-64-23:https  CLOSE_WAIT 5316
  TCP    192.168.1.129:51620    as-40816:https      CLOSE_WAIT 5316
  TCP    192.168.1.129:51633    a23-13-68-236:http  CLOSE_WAIT 5316
  TCP    192.168.1.129:52034    outlook:https       ESTABLISHED 3580
  TCP    192.168.1.129:52035    outlook:https       ESTABLISHED 3580
  TCP    192.168.1.129:52041    outlook:https       ESTABLISHED 3580
  TCP    192.168.1.129:52045    outlook:https       ESTABLISHED 3580
  TCP    192.168.1.129:52157    outlook:https       ESTABLISHED 3580
  TCP    192.168.1.129:52159    outlook:https       ESTABLISHED 3580
  TCP    192.168.1.129:52238    65.52.209.62:https   TIME_WAIT   0
  TCP    [2601:8:1840:363:4029:e85:784d:40f2]:52039  bn1wns2011603:https ESTABLISHED 2924
  TCP    [2601:8:1840:363:4029:e85:784d:40f2]:52114  bn1wns2011717:https ESTABLISHED 2924

C:\>
```

Other Useful Commands

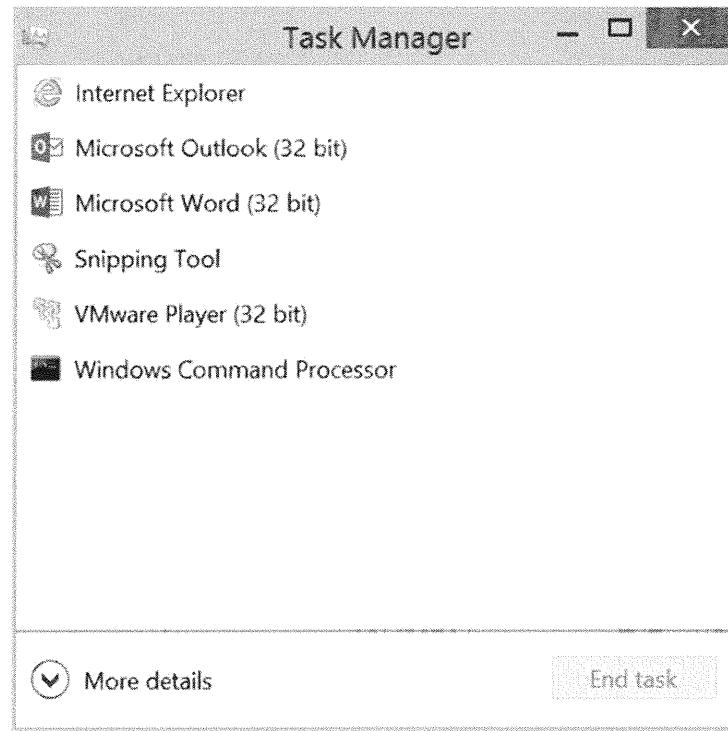
Some other commands you can use at the cmd prompt are the following:

- **cls**: Clears everything on the screen and returns you to the top of the cmd window
- **dir**: Displays a directory listing
- **cd **: Returns you to c:\ from whatever directory you are in

Task Manager

Another useful built-in tool in Windows is the Task Manager. To open and use the Task Manager, perform the following steps:

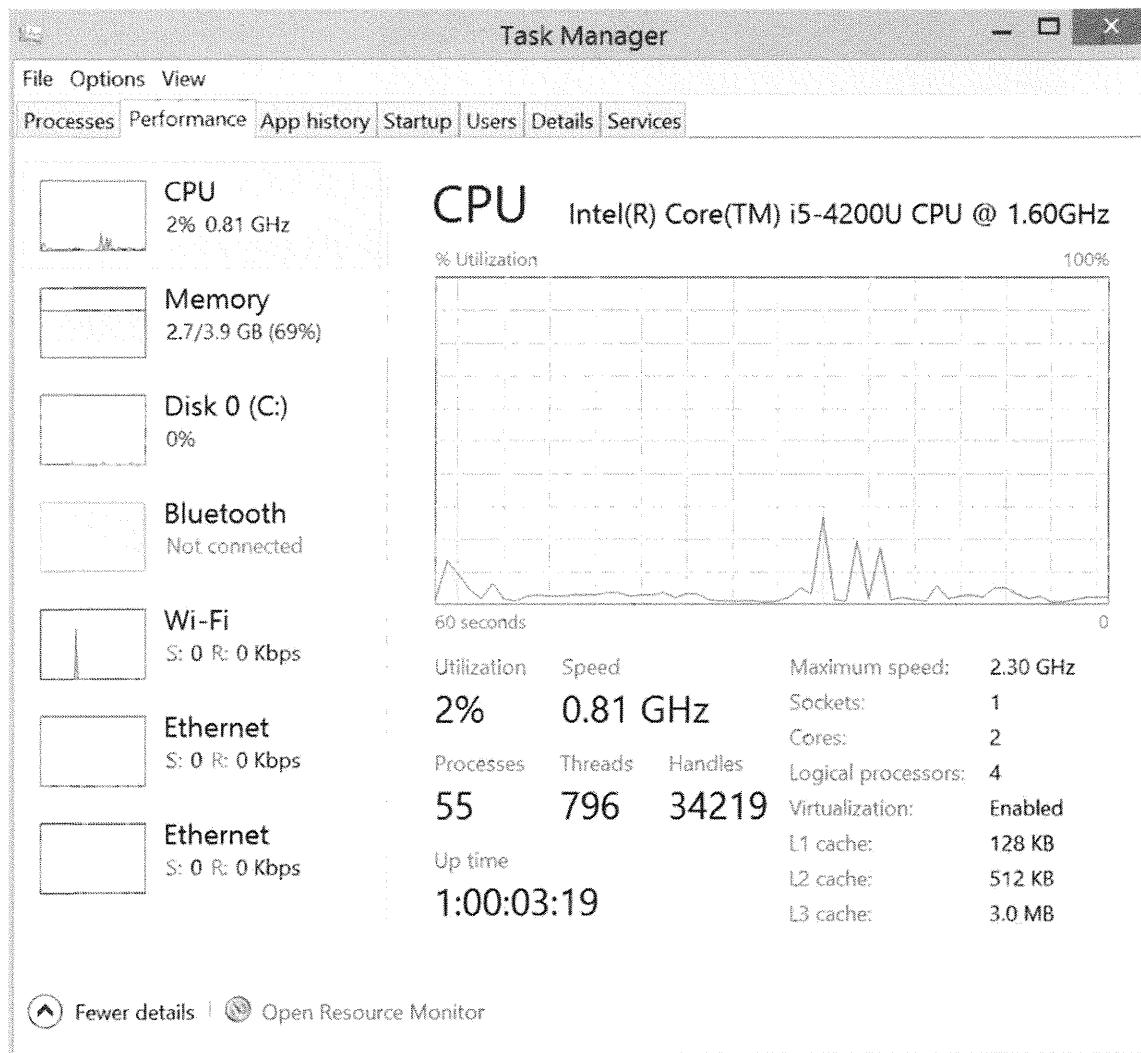
1. To open the Task Manager, hold down *CTRL-ATL-DEL* at the same time and after the menu displays, click *Task Manager*. The default screen that opens shows which applications are running on the system.



2. To show additional details, at the bottom of the window, click *More details*.

Task Manager						
File Options View		Processes	Performance	App history	Startup	Users
Name	Status	5%	68%	0%	0%	
Apps (8)						
> Internet Explorer (2)		0%	96.3 MB	0 MB/s	0 Mbps	
> Microsoft Outlook (32 bit)		0%	47.3 MB	0 MB/s	0 Mbps	
> Microsoft Word (32 bit)		0%	170.6 MB	0 MB/s	0 Mbps	
> Snipping Tool		2.0%	2.1 MB	0 MB/s	0 Mbps	
> Task Manager		0.5%	8.2 MB	0 MB/s	0 Mbps	
> VMware Player (32 bit)		0%	16.2 MB	0 MB/s	0 Mbps	
> Windows Command Processor		0%	1.1 MB	0 MB/s	0 Mbps	
> Windows Explorer (4)		0.6%	47.1 MB	0 MB/s	0 Mbps	
Background processes (22)						
, COM Surrogate		0%	2.2 MB	0 MB/s	0 Mbps	
Device Association Framework ...		0%	5.5 MB	0 MB/s	0 Mbps	
Host Process for Setting Synchronizer...		0%	0.9 MB	0 MB/s	0 Mbps	
Host Process for Windows Tasks		0%	3.5 MB	0 MB/s	0 Mbps	
(Fewer details		End task				

3. From this window, you can check the running processes on the device, the performance trends, and the applications that are currently running. This tool is helpful when an application stops responding. You can open Task Manager, highlight the application, and then close the offending application. By clicking the *Performance* tab, you can see CPU and memory usage.



There are many other Windows functions that are not covered in this book. Our goal is to give you the basics, so that you can quickly install and run the tools covered throughout the following chapters.

Setting Up the Directory Structure

Now that you have installed the operating system and worked with some of the tools that are built into it, you need to set up your directory structure, so that it's consistent with the directory structure used for installing and storing the tools discussed in this book. To set up the directory structure, perform the following steps:

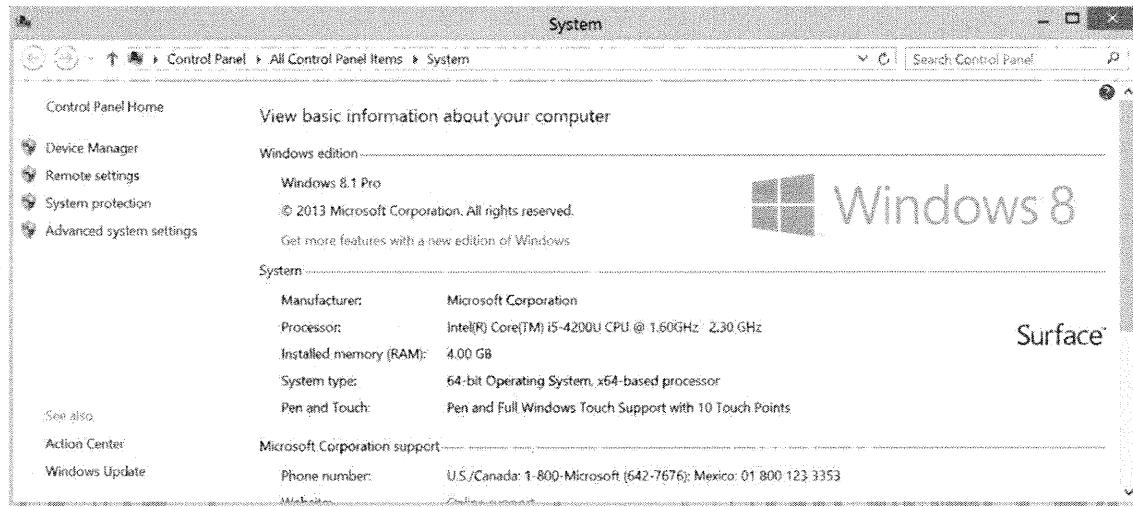
1. To display the Windows 8 traditional desktop, from the Start area, click *Desktop*. Right-click the desktop, click *New*, and then click *Folder* to create a new folder.



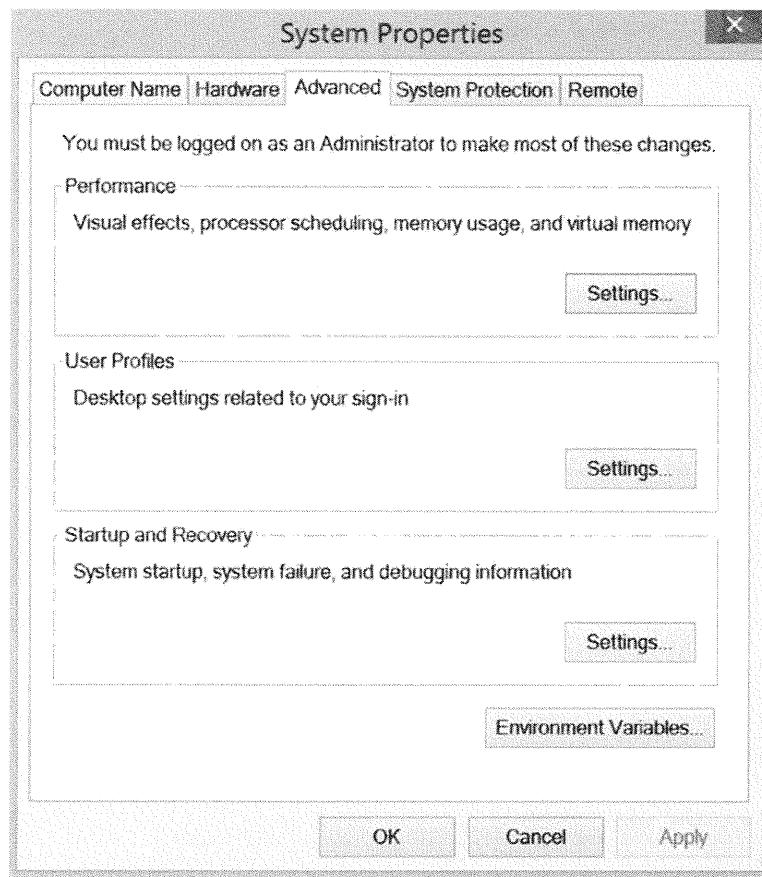
2. After a new folder displays on the desktop, name the folder **tools**.
3. The exercises in this book require you to run several tools from the command line. Therefore, you need to add the new folder you created, c:\tools, to PATH. If you do this, you won't have to navigate to the tools folder each time you want to run an application. In addition, to install Java in future exercises, you need to add Java to your path in later exercises. To add the folder to PATH, in the lower left-hand side (where the Start menu was previously located), right-click the Windows icon (which looks like the following):



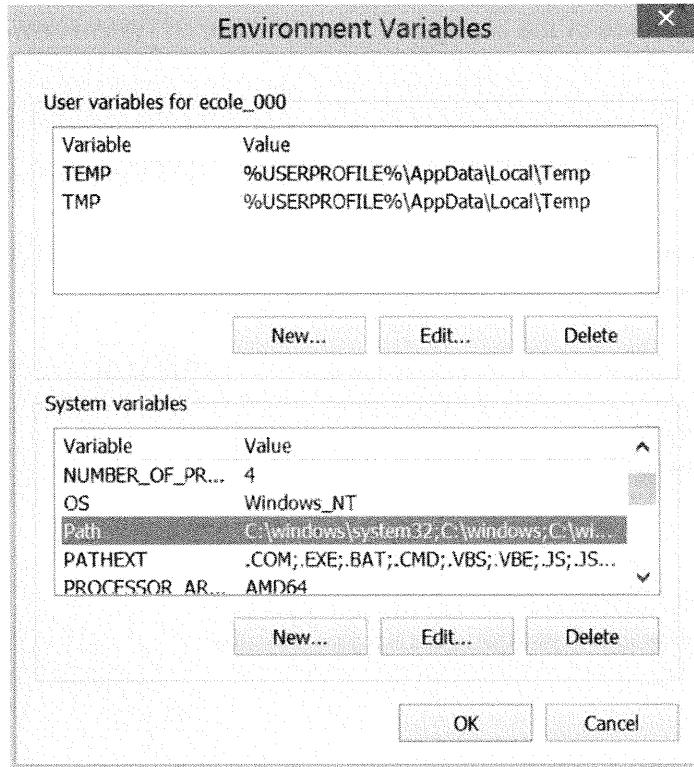
4. From the menu, click *System*.



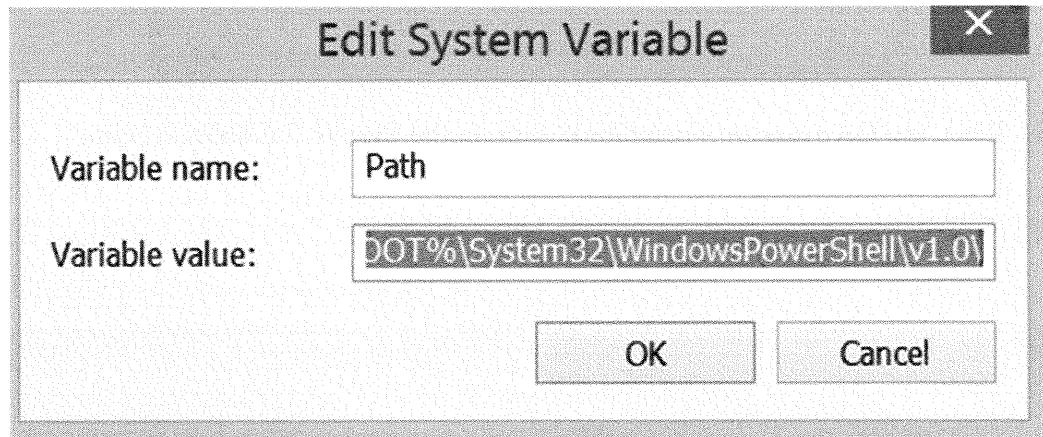
5. On the left-hand side of the dialog box, click *Advanced system settings*.



6. Click the *Environment Variables* button. In the System variables section, highlight the *Path* line.



- With the Path line highlighted, click *Edit*.



- In the Variable value field, move your cursor to the end of the line and type the following text exactly as it is shown here:

;c:\tools

Click *OK* on each of the Edit System Variable, Environment Variables, and System Properties windows.

You can now run the executables located in c:\tools from any directory in your file structure. This saves a lot of time when you are using the command prompt and want to run an application from it.

Linux (1)

- Learn how to log in
- Create accounts
- Understand file and directory manipulation and the associated commands, which include:
 - ls
 - ls -al
 - mkdir
- Learn how to use the power of man
- Change directories using the following:
 - cd
 - pwd

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Linux (1)

Linux is an open source operating system that runs on a wide range of hardware platforms. So what is open source? As an open source system, Linux is protected under the GNU General Public License, which guarantees the freedom to use and change the software it covers. Numerous Linux distributions are available from many companies, and each distribution has its own advantages and disadvantages. With these characteristics comes a faithful user following who think that their preferred distribution is the best. Some of the Linux distributions that are currently available include Red Hat, SUSE, Debian, and Mandrake.

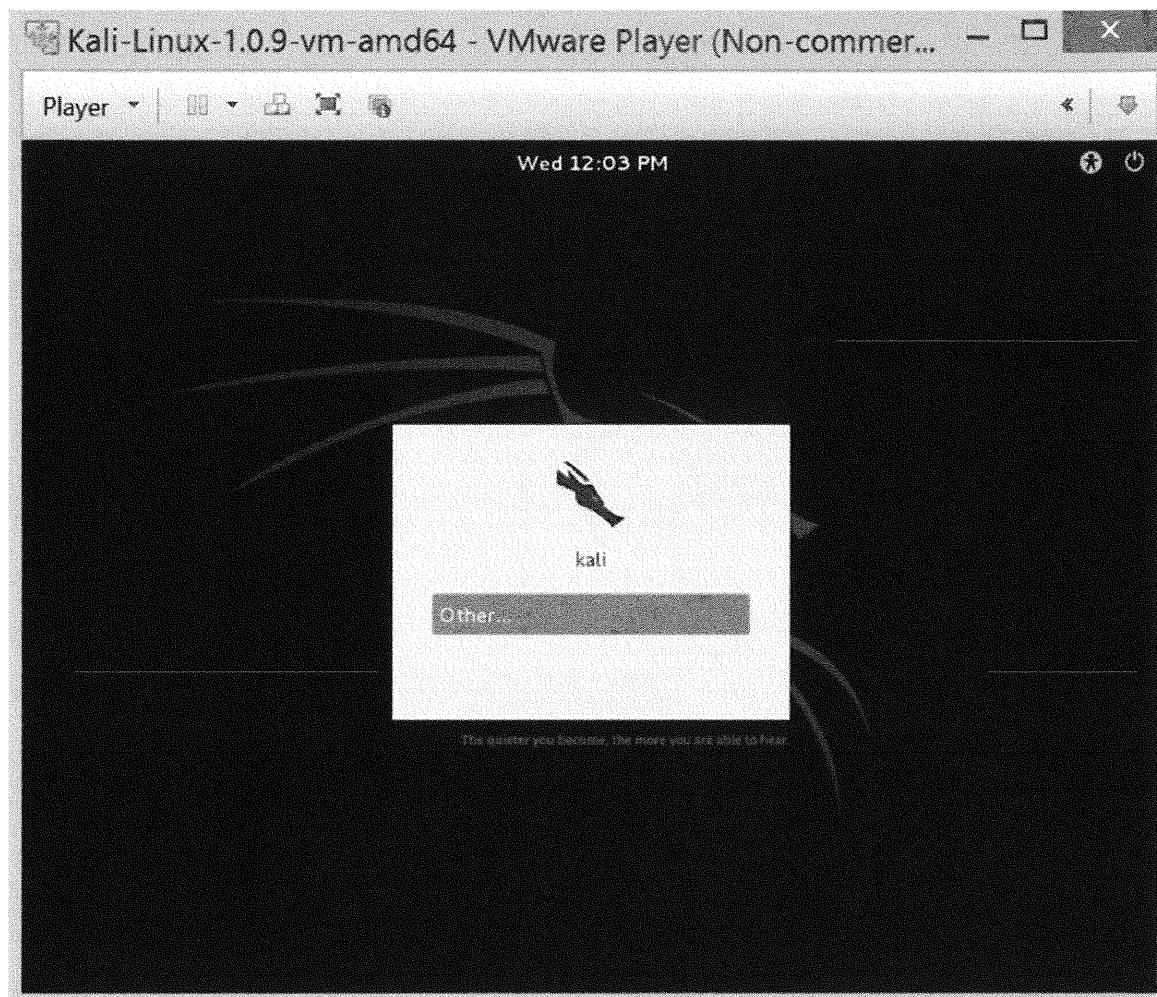
As Linux became popular, various versions were created. The version we use in class is Kali Linux.

At the heart of each distribution is the kernel, which interacts directly with the hardware. The kernel handles such functions as memory management, security, and resource allocation. The kernel also provides features such as true multitasking, threading, and TCP/IP networking. Contrary to popular belief, the kernel is, in fact, Linux. All other applications and programs are part of a particular distribution.

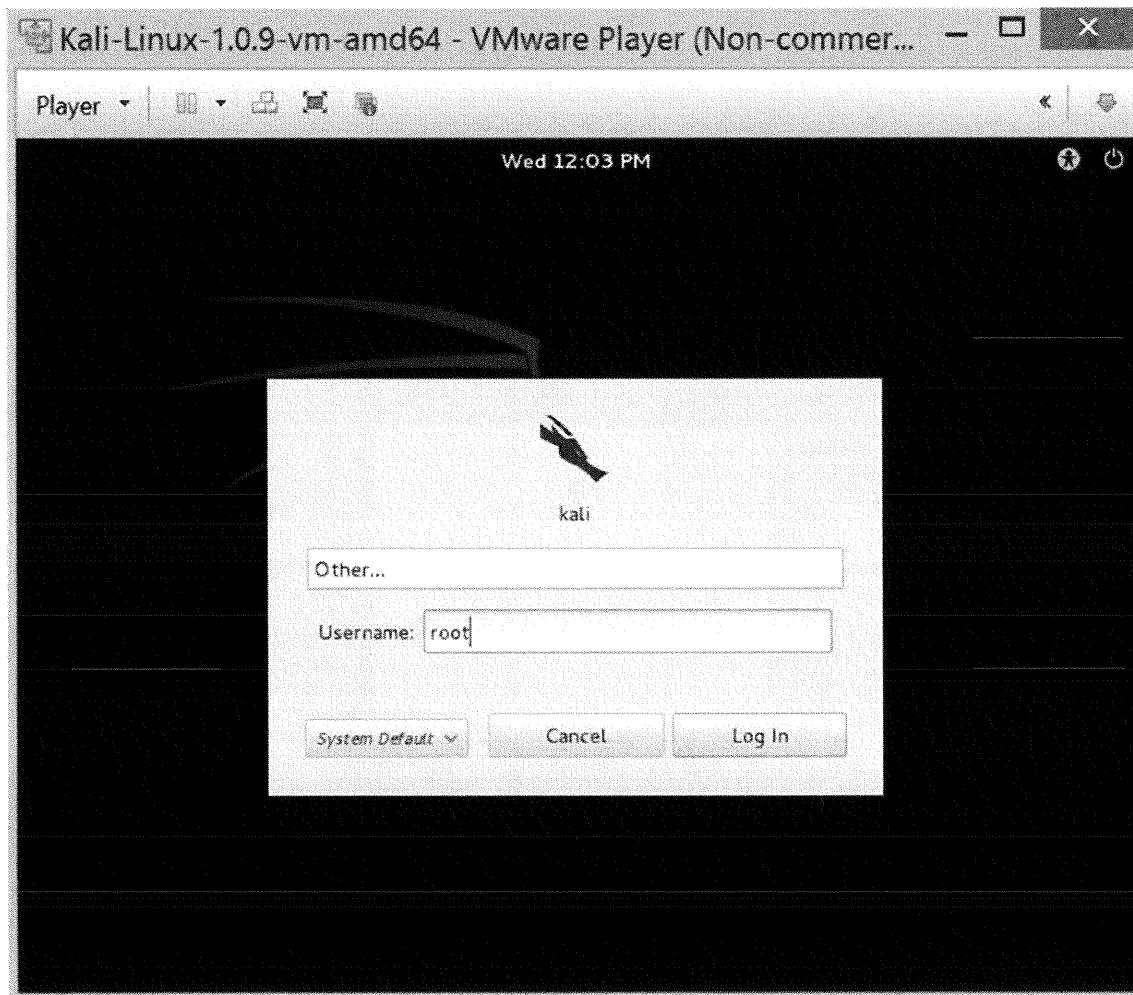
The Linux shell is another name for the command shell, which is similar in function to a DOS shell. It is the program that gives you an interface to type commands, and it accepts the commands you type. For the following examples, remember that nearly everything in Linux is case sensitive.

Starting Up Kali

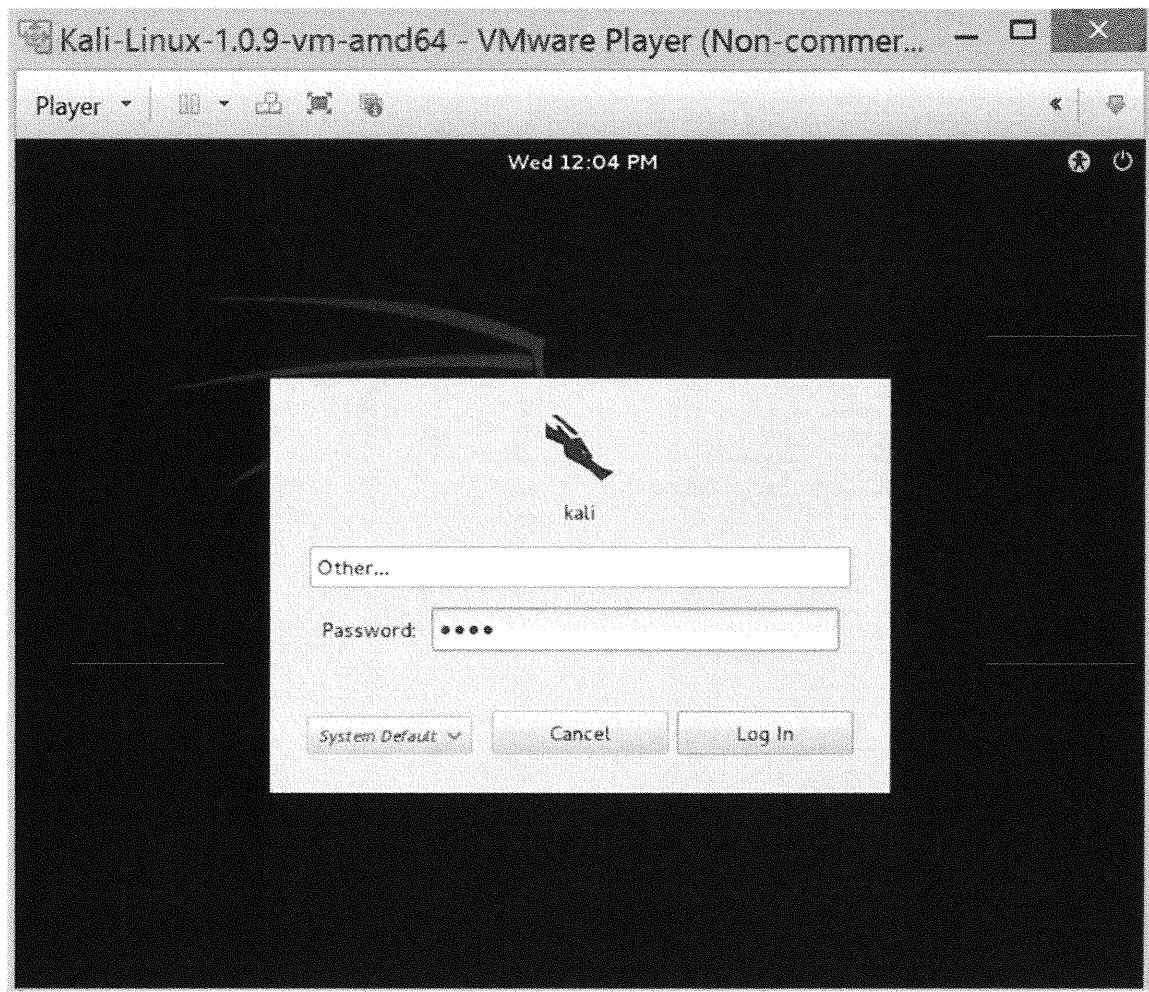
After the system boots, you are required to log on to the system. Log on with a user ID of **root** and a password of **toor** (remember the password is just root backwards) and press **Enter**. The following shows the specific steps for logging into the system.



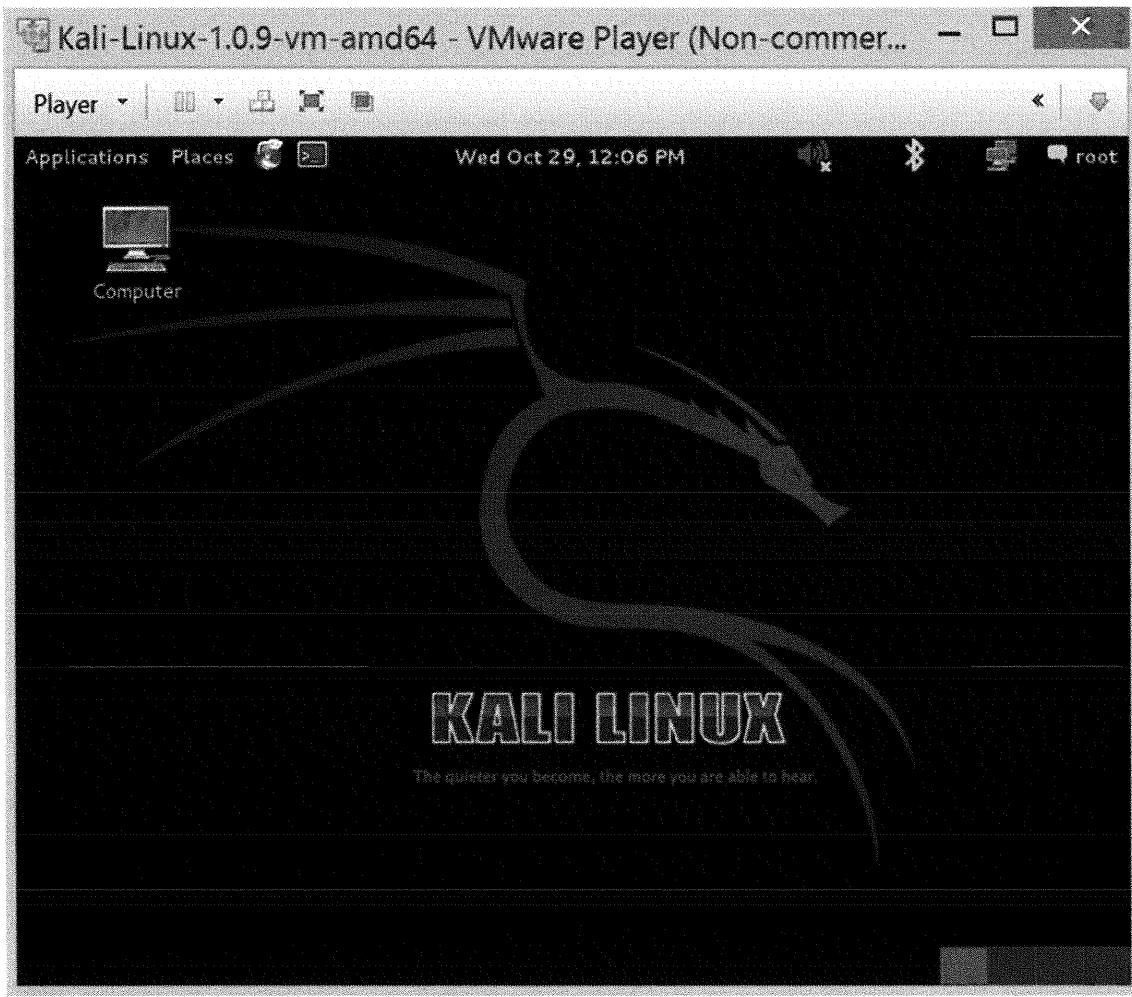
1. After Kali starts, click *Other*, type **root** as the username, and click *Log In*.



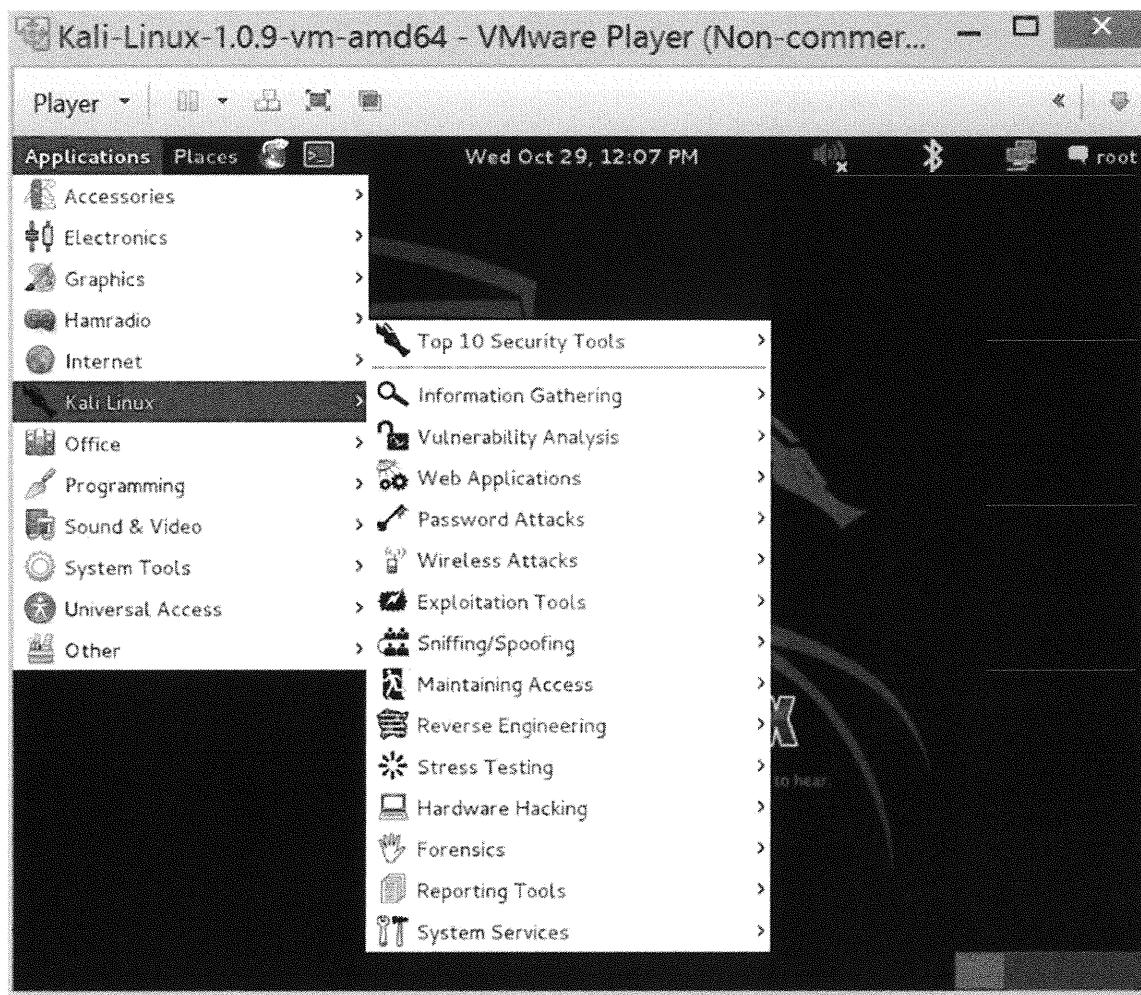
-
2. When prompted for a password, type **toor** (which is root backwards) and click *Log In*.



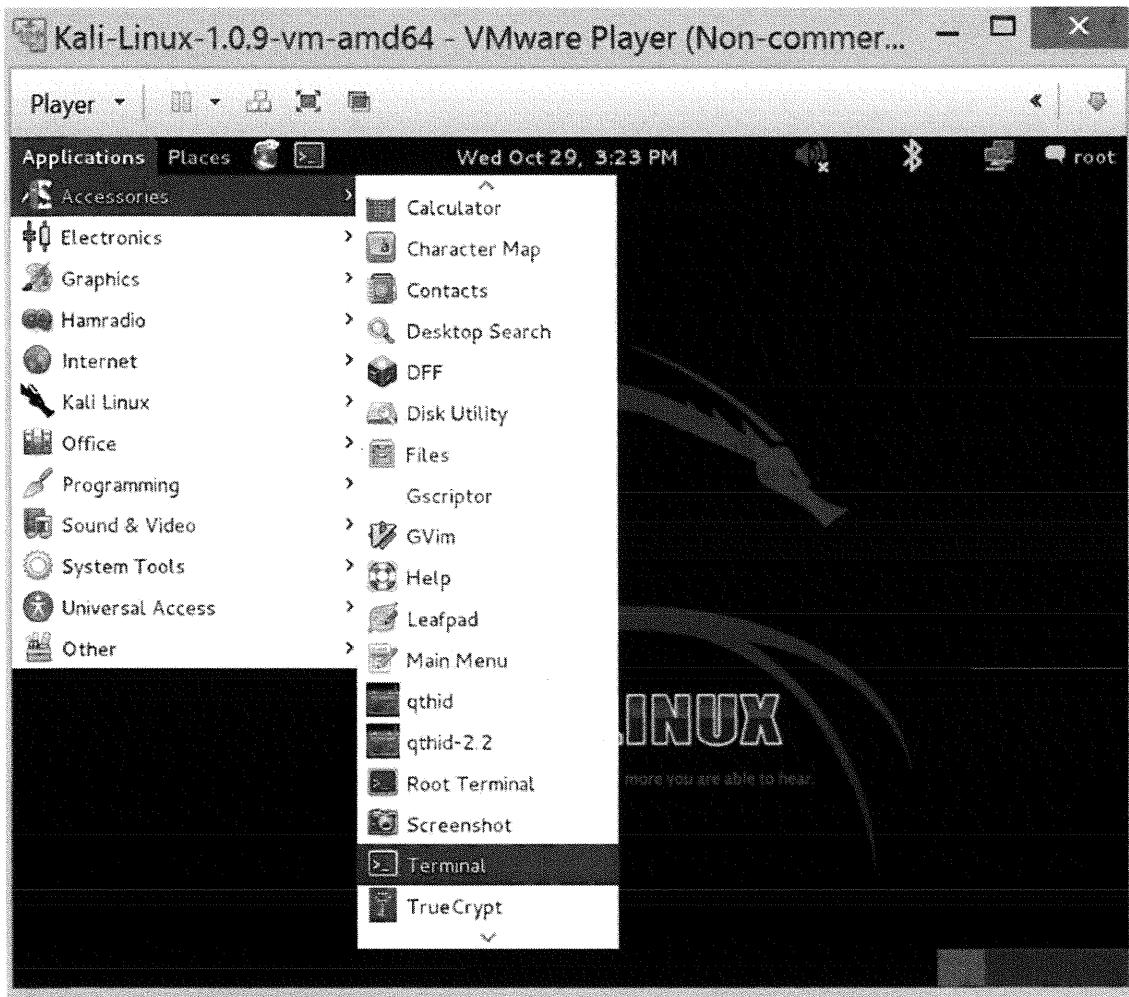
3. Now, Kali Linux starts and you are ready to run the labs.



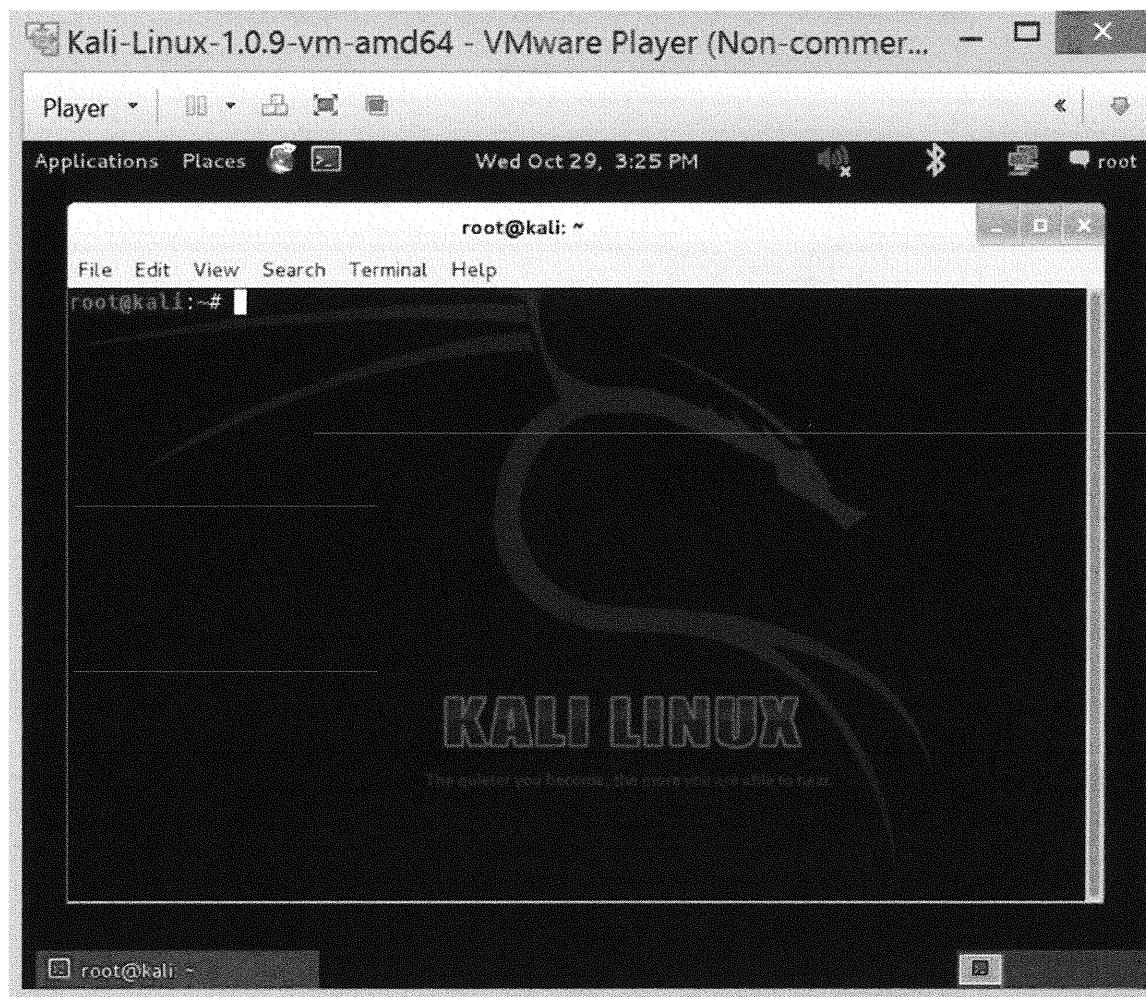
4. In the upper right menu, click *Applications* -> *Kali Linux* to view all the tools that are installed.



5. Many of the tools are run from a root terminal shell. Although there are many ways to open a root terminal shell, the easiest way is to go to the *Applications* menu, click *Accessories*, and click *Terminal*.



6. A root terminal shell opens. You can tell you are at a root shell because the prompt is the # symbol. If the prompt is a \$ sign, then you are logged in as a normal user. In a typical production environment, you always log in as a normal user. To run many of the labs in this course, you will log in as root.



7. To learn about the tools and ensure that everything works correctly, we will work from a root shell in the labs.

Linux (2)

- File viewing/manipulation:
 - ls
 - less
- Accounts:
 - su
 - whoami
- System configuration:
 - ping
 - netstat
 - ps

SANS Security Essentials - © 2016 Secure Anchor Consulting LLC

Linux (2)

This section introduces you to the basics of Linux by covering some of the most common commands, files, and directories used in Linux. Each topic includes a brief description and an example of how the topic is used. You can find more information on each topic by typing **man**, which stands for “manual.” For example, issue the following command at a shell prompt:

man man

This command displays a manual that describes the **man** command and also demonstrates how man pages are formatted.

The screenshot shows a terminal window titled "root@kali: ~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The title bar says "MAN(1) Manual pager utils MAN(1)". The main content area displays the man page for "pager utils".

```
man - an interface to the on-line reference manuals

NAME
    man - an interface to the on-line reference manuals

SYNOPSIS
    man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
    locale] [-m system[,...]] [-M path] [-S list] [-e extension] [-i|-I]
    [--regex|--wildcard] [--names-only] [-a] [-u] [--no-subpages] [-P
    pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justifi-
    cation] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
    [[section] page ...] ...
    man -k [apropos options] regexp ...
    man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
    man -f [whatis options] page ...
    man -l [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
    locale] [-P pager] [-r prompt] [-7] [-E encoding] [-p string] [-t]
    [-T[device]] [-H[browser]] [-X[dpi]] [-Z] file ...
    man -w|-W [-C file] [-d] [-D] page ...
    man -c [-C file] [-d] [-D] page ...
    man [-hV]

DESCRIPTION
    | Manual page man(1) line 1 (press h for help or q to quit) |
```

After you are done viewing a man page, type **q** to exit.

The ls Command

One of the most basic commands in Linux is **ls**. Just as you use **dir** in DOS, type **ls** to output a listing of the directories and files that are contained in the current directory. Open a root shell as described previously. Now, in the command shell, type **ls** (as shown in the following screen). You can also type **ls -l** to get a full listing, including permissions.

```

root@kali:~# ls
Desktop
root@kali:~#
root@kali:~# cd /
root@kali:/#
root@kali:/# ls
9 etc lib media root srv var
bin example.conf.json lib32 mnt run sys vmlinuz
boot home lib64 opt sbin
dev initrd.img lost+found proc selinux usr
root@kali:/#
root@kali:/# ls -l
total 96
-rw-r--r-- 1 root root 0 Aug 10 06:42 0
drwxr-xr-x 2 root root 4096 Aug 12 04:48 bin
drwxr-xr-x 4 root root 4096 Aug 26 04:16 boot
drwxr-xr-x 15 root root 3340 Oct 29 12:00 dev
drwxr-xr-x 182 root root 12288 Oct 29 15:27 etc
-rw-r--r-- 1 root root 625 May 10 10:10 example.conf.json
drwxr-xr-x 2 root root 4096 Jul 21 11:26 home
lrwxrwxrwx 1 root root 33 Aug 12 03:53 initrd.img -> /boot/initrd.img-3.14-
kalil-amd64
drwxr-xr-x 17 root root 4096 Aug 12 03:53 lib
drwxr-xr-x 2 root root 4096 Aug 12 03:53 lib32

```

As with most commands in Linux, you can specify options to change the result of the command's execution. For example, type **ls -al** in the command shell (see the following screen).

```

root@kali:~#
root@kali:~#
root@kali:/# ls -al
total 104
drwxr-xr-x 24 root root 4096 Aug 12 04:00 .
drwxr-xr-x 24 root root 4096 Aug 12 04:00 ..
-rw-r--r-- 1 root root 0 Aug 10 06:42 0
drwxr-xr-x 2 root root 4096 Aug 12 04:48 bin
drwxr-xr-x 4 root root 4096 Aug 26 04:16 boot
drwxr-xr-x 15 root root 3340 Oct 29 12:00 dev
drwxr-xr-x 182 root root 12288 Oct 29 15:27 etc
-rw-r--r-- 1 root root 625 May 10 10:10 example.conf.json
drwxr-xr-x 2 root root 4096 Jul 21 11:26 home
lrwxrwxrwx 1 root root 33 Aug 12 03:53 initrd.img -> /boot/initrd.img-3.14-
kalil-amd64
drwxr-xr-x 17 root root 4096 Aug 12 03:53 lib
drwxr-xr-x 2 root root 4096 Aug 12 03:53 lib32
drwxr-xr-x 2 root root 4096 Aug 12 03:53 lib64
drwxr----- 2 root root 16384 Aug 12 03:53 lost+found
drwxr-xr-x 3 root root 4096 Aug 10 06:13 media
drwxr-xr-x 3 root root 4096 Aug 26 04:15 mnt
drwxr-xr-x 5 root root 4096 Aug 12 03:53 opt
dr-xr-xr-x 134 root root 0 Oct 29 12:00 proc
drwxr-xr-x 12 root root 4096 Oct 29 15:28 root

```

The **-a** option tells the command interpreter to show all files, and **-l** tells it to use the long listing format. These are two of the many options that you can use with **ls**.

Changing Directories and Creating Directories

Now that you can tell what files and directories the root directory contains, let's move back to the **/** directory. To change directories, you use the **cd** command; here, you can use the **cd /** command, as shown in the following screen. You can also type **pwd** (print working directory) at any time to print the working directory you are currently in.

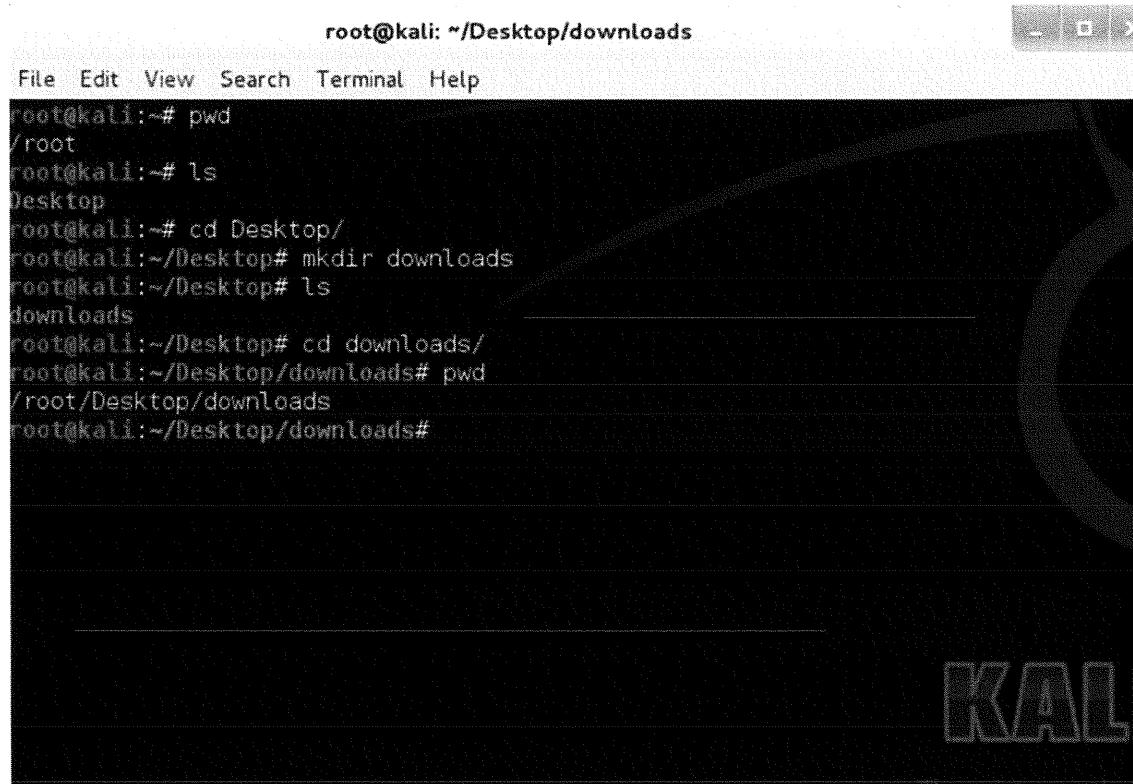


A screenshot of a terminal window titled "root@kali: /". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area shows the following command history:

```
root@kali:~# pwd
/root
root@kali:~#
root@kali:~# cd /
root@kali:/#
root@kali:/# pwd
/
root@kali:/#
```

You can use the **mkdir** command to create a directory. The format of the **mkdir** command is **mkdir<new directory name>**. For example, type **mkdir downloads** to create a location where to save files that have been downloaded from the Internet.

Issuing **ls** after the **mkdir** command shows the newly created download directory, as shown in the previous screen.



The screenshot shows a terminal window titled "root@kali: ~/Desktop/downloads". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content is as follows:

```
root@kali:~/Desktop/downloads
File Edit View Search Terminal Help
root@kali:~# pwd
/root
root@kali:~# ls
Desktop
root@kali:~# cd Desktop/
root@kali:~/Desktop# mkdir downloads
root@kali:~/Desktop# ls
downloads
root@kali:~/Desktop# cd downloads/
root@kali:~/Desktop/downloads# pwd
/root/Desktop/downloads
root@kali:~/Desktop/downloads#
```

Determining Directory Placement

After using **cd** and **ls** to learn about the Linux structure of changing directories, you might not realize which directory you are currently in. You can determine where you are in the directory structure by typing **pwd**.

A screenshot of a terminal window titled "root@kali: /bin". The window has a dark background with light-colored text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu, the command prompt is "root@kali:~#". The user runs several commands: "pwd" shows the current directory is "/root"; "cd /" changes it to the root directory; "ls" lists files and directories including "etc", "bin", "dev", "home", "lib", "media", "mnt", "opt", "run", "sbin", "selinux", "sys", "var", and "vmlinuz". Then, "cd bin" changes to the "/bin" directory, and "ls" lists its contents: "bash", "domainname", "ls", "ntfsinfo", "tailf", "bunzip2", "echo", "lsblk", "ntfscls", "tar", "busybox", "ed", "lsmod", "ntfsmftalloc", "tempfile", "bzcat", "egrep", "mkdir", "ntfsmove", "touch", "bzcmp", "false", "mknode", "ntfstruncate", "true", "bzdiff", "fgconsole", "mktemp", "ntfswipe", "unlockmgr", "server", "bzegrep", "fgrep", "more", "pidof", "bzexe", "findmnt", "mount", "ps", "uname", "bzfgrep", "fuser", "mountpoint", "ptx", "uncompress", "bzgrep", "fusermount", "mt", "usleep". The "mount" command is highlighted with a red box.

```
root@kali:~# pwd
/root
root@kali:~# cd /
root@kali:/#
/
root@kali:/# ls
etc      lib      media   root    srv    var
bin     example.conf.json lib32    mnt    run    sys    vmlinuz
boot    home    lib64    opt    sbin
dev     initrd.img  lost+found proc    selinux  usr
root@kali:/# cd bin
root@kali:/bin# pwd
/bin
root@kali:/bin# ls
bash      domainname  ls        ntfsinfo  tailf
bunzip2   echo        lsblk    ntfscls  tar
busybox   ed          lsmod   ntfsmftalloc tempfile
bzcat    egrep        mkdir   ntfsmove touch
bzcmp    false        mknode  ntfstruncate true
bzdiff   fgconsole   mktemp  ntfswipe unlockmgr
server
bzegrep  fgrep       more    pidof
bzexe   findmnt    mount   ps
bzfgrep  fuser      mountpoint ptx
bzgrep   fusermount  mt      usleep
root@kali:/bin#
```

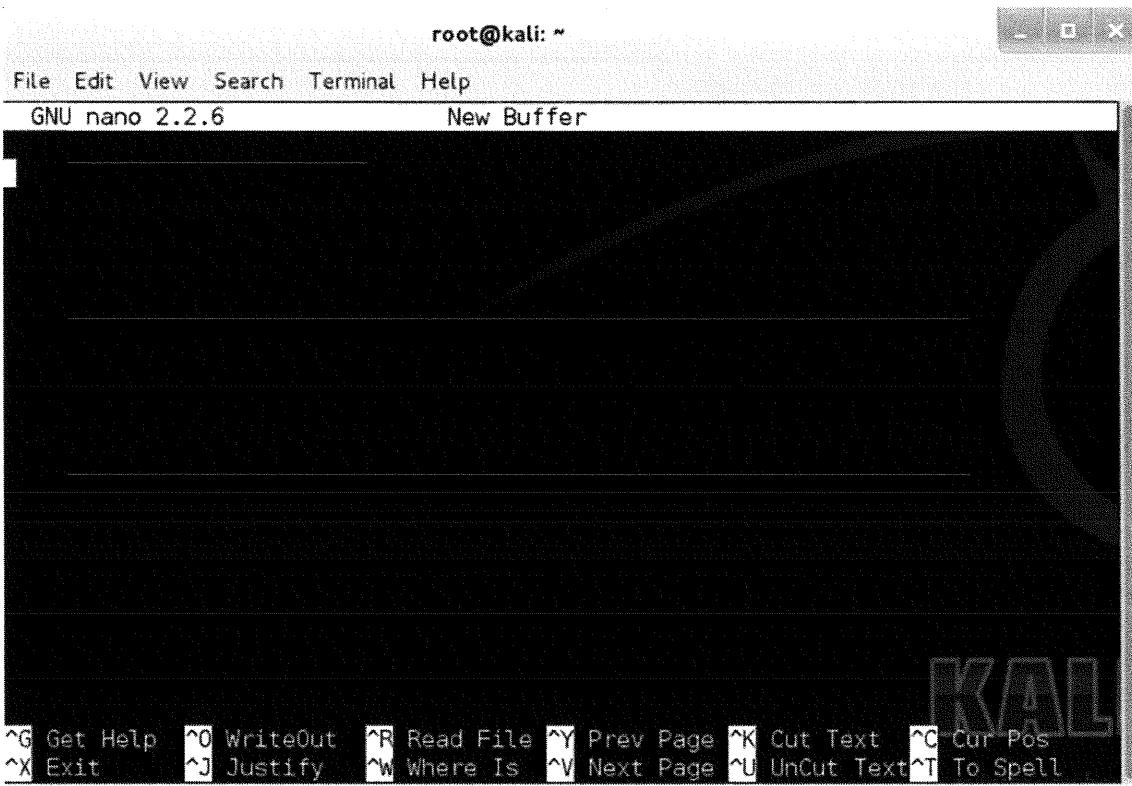
Creating Files

To create a file, you can use nano. To run nano, open a terminal window, confirm you are in the root directory by typing **pwd**, and then type **nano**.

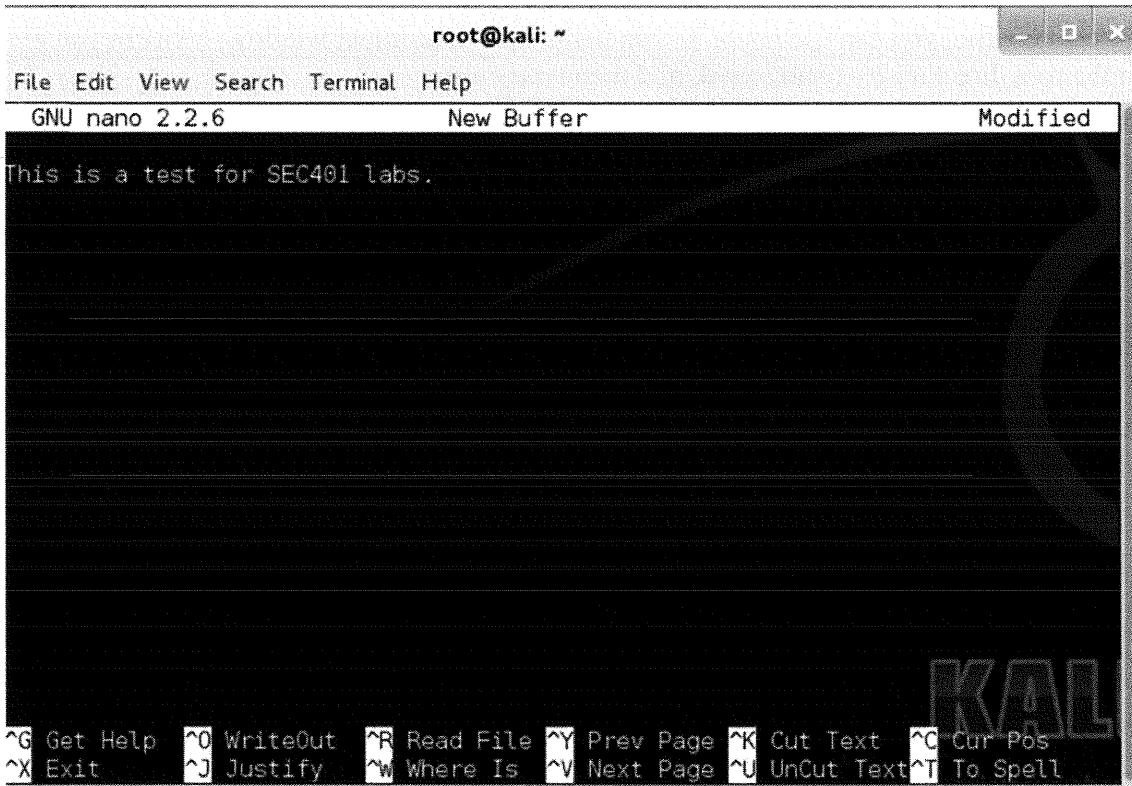
A screenshot of a terminal window titled "root@kali: ~". The window has a dark background with light-colored text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu, the command prompt is "root@kali:~#". The user runs "pwd" to show they are in the root directory, then types "nano" to open the nano text editor. The "nano" command is highlighted with a red box.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# pwd
/root
root@kali:~# nano
```

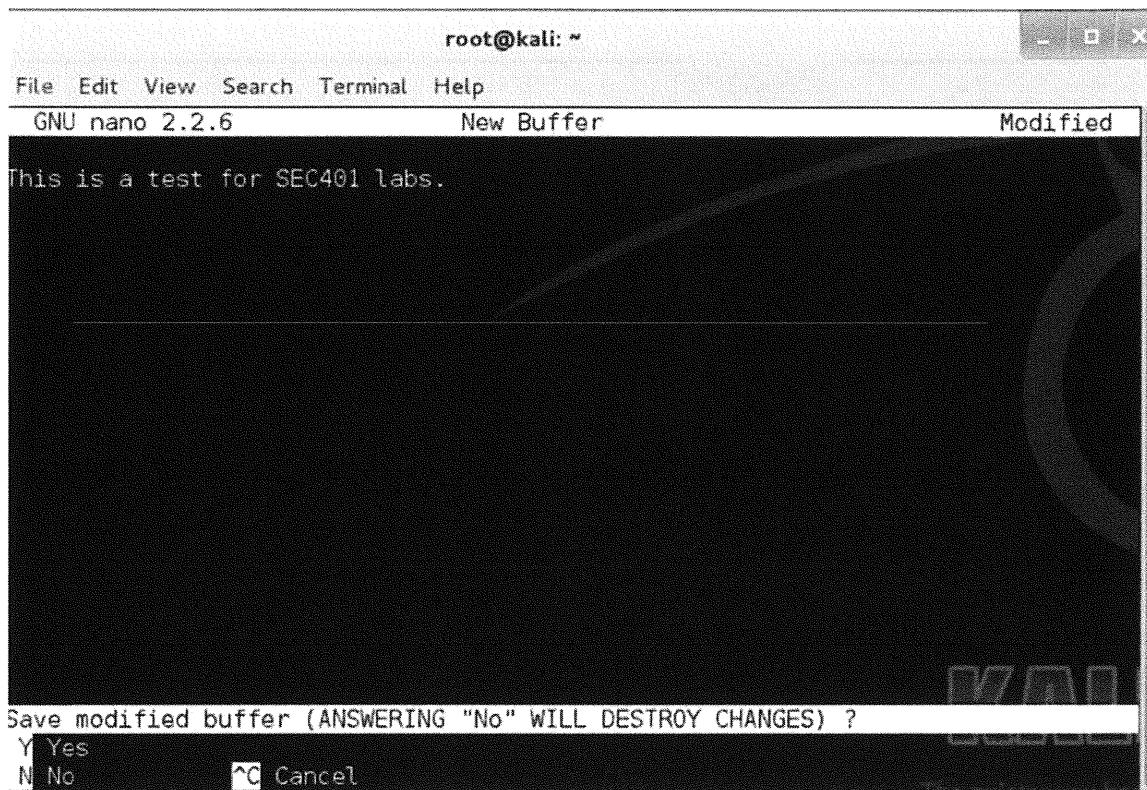
The nano text editor displays.



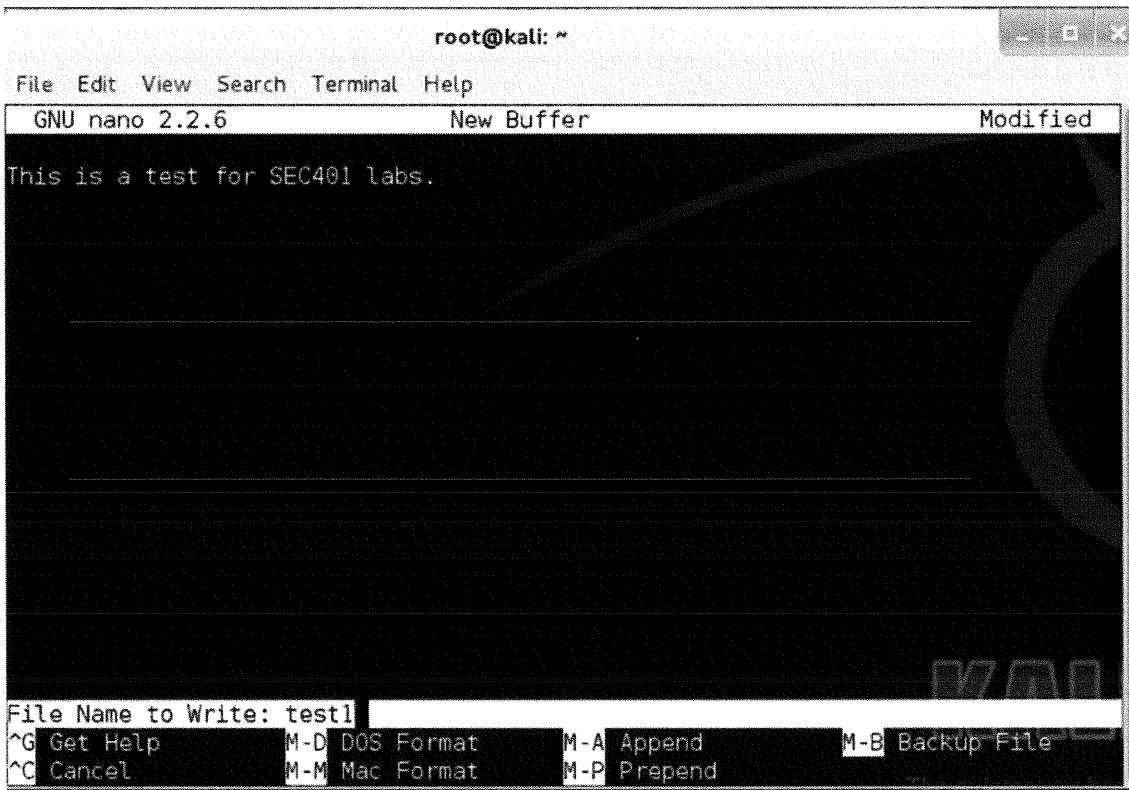
After the text editor opens, type a phrase in the editor.



After typing your phrase, hold down the **CTRL** key and type **X**. In the Save menu, type **Y** to save the file.



Type the name of the file **test1**, and press *Enter* to save the file and exit the program.



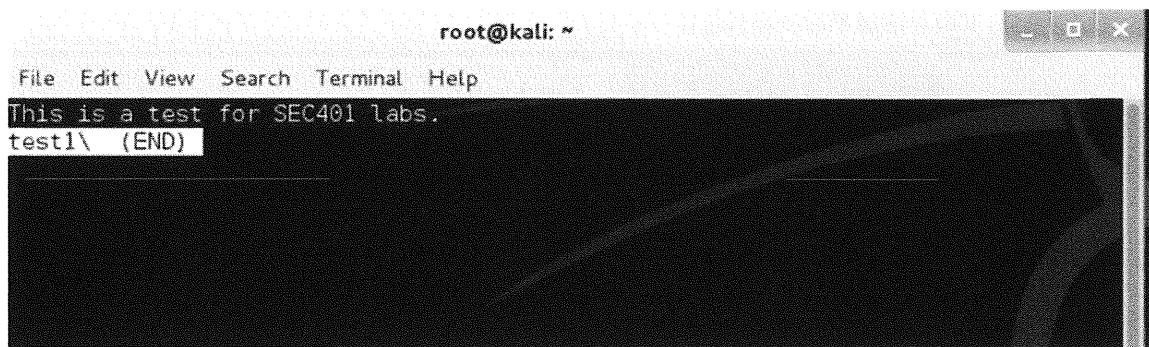
Viewing Files

Although there are many ways to view the contents of a file in Linux, one command you can use for this function is less. To view the contents of the `linux_lab` file that you just created, perform the following steps:

1. Open a root terminal window.
2. Issue the `cd /root` command to change to the root directory.
3. Type `ls` to confirm that your file is listed.

```
root@kali:~# cd /root
root@kali:~/root# pwd
/root
root@kali:~/root# ls
Desktop test1
root@kali:~/root# less test1
```

4. Type `less test1` to view the content of the file.



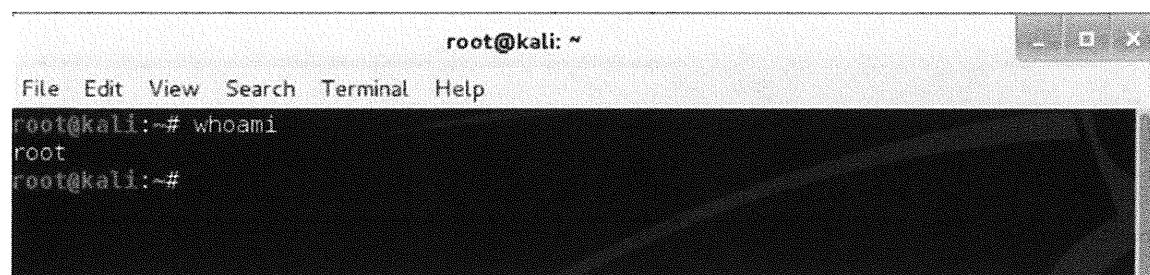
A screenshot of a terminal window titled "root@kali: ~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The title bar also displays "root@kali: ~". The main area of the terminal shows the following text:
This is a test for SEC401 labs.
test1\ (END)

5. Use the arrow keys to navigate through the contents of the file. After you finish, type q to exit and return back to the command shell.

Determining Account Types

As you gain more Linux experience, you will find yourself ssh-ing (SSH is secure shell) to other systems on your network or on the Internet. To determine the account in which you are logged in, type the following:

whoami



A screenshot of a terminal window titled "root@kali: ~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The title bar also displays "root@kali: ~". The main area of the terminal shows the following text:
root@kali:~# whoami
root
root@kali:~#

Common Files and Directories

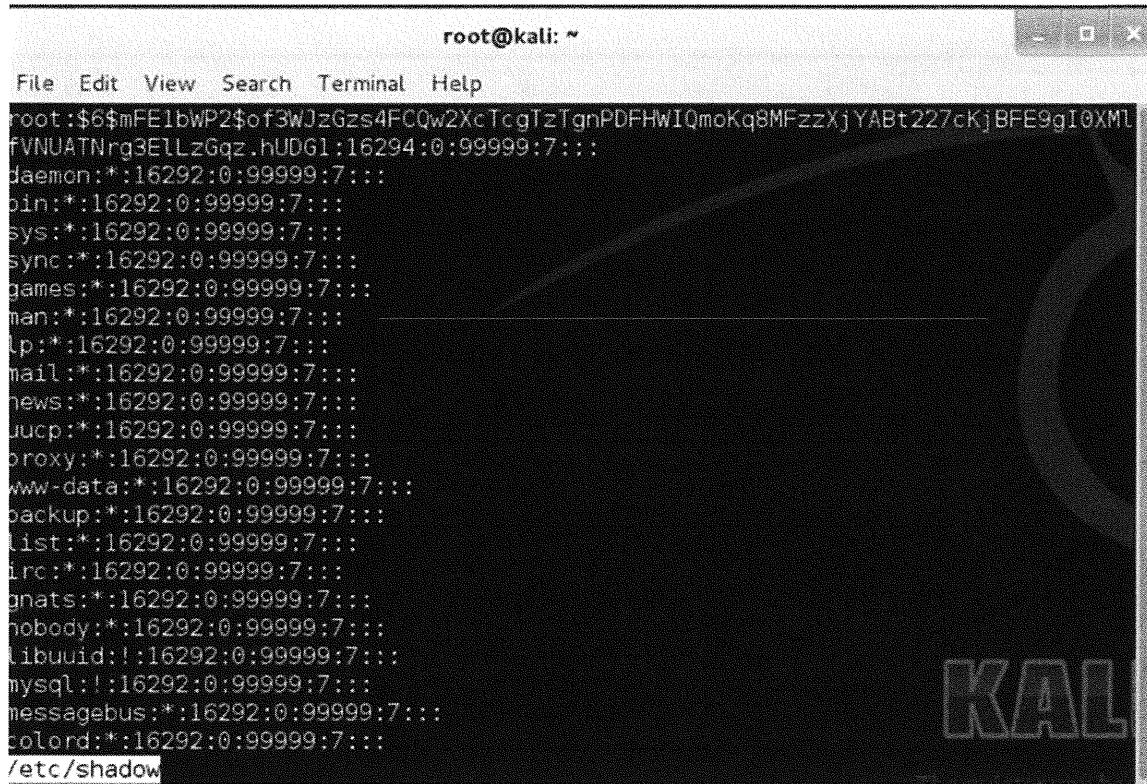
Some user information is stored in a file called “passwd,” which is located in the /etc directory. This file also contains the path to the user's home directory, as well as to the current shell. Issue the **less /etc/passwd** command to view the contents of passwd. Type **q** to exit less.



The screenshot shows a terminal window titled "root@kali: ~". The window contains the output of the "less /etc/passwd" command. The text lists various system users and their details, such as their user ID (UID), group ID (GID), home directory, and shell. The terminal has a dark background with light-colored text. The Kali Linux logo is visible in the bottom right corner of the window frame.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/false
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
mysql:x:101:103:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
colord:x:103:107:colord colour management daemon,,,:/var/lib/colord:/bin/false
usbmux:x:104:46:usbmux-daemon,,,:/home/usbmux:/bin/false
/etc/passwd
```

Because the Linux default is to use a shadow file, the password for all accounts is listed as x. The shadow file contains an encrypted version of the actual password and is used to enhance security. The permissions on the shadow file are generally more restrictive than the passwd file. To view the contents of the shadow file, issue the **less /etc/shadow** command. Type **q** to exit less.



```
root@kali: ~
File Edit View Search Terminal Help
root:$6$mfE1bwP2$of3WJzGzs4FCQw2XcTcgTzTgnPDFHWI0moKq8MFzzXjYAbt227cKjBfE9gI0XMT
fVNUATNrg3ElLzGqz.hUDG1:16294:0:99999:7:::
daemon:*:16292:0:99999:7:::
bin:*:16292:0:99999:7:::
sys:*:16292:0:99999:7:::
sync:*:16292:0:99999:7:::
games:*:16292:0:99999:7:::
man:*:16292:0:99999:7:::
lp:*:16292:0:99999:7:::
mail:*:16292:0:99999:7:::
news:*:16292:0:99999:7:::
uucp:*:16292:0:99999:7:::
proxy:*:16292:0:99999:7:::
www-data:*:16292:0:99999:7:::
backup:*:16292:0:99999:7:::
list:*:16292:0:99999:7:::
irc:*:16292:0:99999:7:::
gnats:*:16292:0:99999:7:::
nobody:*:16292:0:99999:7:::
libuuid!:16292:0:99999:7:::
mysql!:16292:0:99999:7:::
messagebus!:16292:0:99999:7:::
colord!:16292:0:99999:7:::
/etc/shadow
```

The shadow file is accessible only by root. If you are not logged in as root, you would not be able to see the contents of the file.

Like Windows, Linux uses the hosts file that contains the IP address and associated hostname for a particular device. In a default install of Linux, the hosts file contains only one entry for localhost. The location of the hosts file in Linux is **/etc/hosts**. To view the contents of the hosts file, issue the **less /etc/hosts** command. Type **q** to exit less.

```
root@kali: ~
File Edit View Search Terminal Help
127.0.0.1      localhost
127.0.1.1      kali

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
/etc/hosts (END)
```

Network Configuration

You can view current network configurations by issuing the ifconfig command. With Linux (if you are not connected to a network), you should receive only an entry for 127.0.0.1, which is the loopback address. Type **ifconfig**.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~#
root@kali:~# ifconfig
eth0      Link encap:Ethernet Hwaddr 00:0c:29:17:f9:ce
          inet addr:192.168.195.128 Bcast:192.168.195.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe17:f9ce/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:11800 errors:0 dropped:0 overruns:0 frame:0
            TX packets:5989 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:17341654 (16.5 MiB) TX bytes:382001 (373.0 KiB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:160 errors:0 dropped:0 overruns:0 frame:0
            TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:9600 (9.3 KiB) TX bytes:9600 (9.3 KiB)

root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
```

To verify that TCP/IP is set up correctly, simply ping 127.0.0.1, which validates that the loopback is properly working. Pinging the loopback is used for many of the exercises. Type **ping 127.0.0.1**. To stop ping, type **CTRL-C**.



The watermark features the Kali Linux logo with the text "The harder you become, the more you are able to harm".

```
root@kali:~# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.834 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.133 ms
64 bytes from 127.0.0.1: icmp_req=3 ttl=64 time=0.058 ms
64 bytes from 127.0.0.1: icmp_req=4 ttl=64 time=0.058 ms
64 bytes from 127.0.0.1: icmp_req=5 ttl=64 time=0.061 ms
64 bytes from 127.0.0.1: icmp_req=6 ttl=64 time=0.058 ms
64 bytes from 127.0.0.1: icmp_req=7 ttl=64 time=0.062 ms
64 bytes from 127.0.0.1: icmp_req=8 ttl=64 time=0.061 ms
64 bytes from 127.0.0.1: icmp_req=9 ttl=64 time=0.064 ms
64 bytes from 127.0.0.1: icmp_req=10 ttl=64 time=0.060 ms
^C
--- 127.0.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9002ms
rtt min/avg/max/mdev = 0.058/0.144/0.834/0.231 ms
root@kali:~#
```

Listening Services and Network Connections

It is always important to know what services are listening on your system, as well as what connections have been made. Examples of listening services are sendmail, rpc, and sshd; each of these listens on a specific port or a number of ports. To display the active network connections on your system, issue the **netstat** command. Depending on how your system is configured, you might receive different results.

```

root@kali:~# netstat -an
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type      State         I-Node Path
unix  14      [ ]        DGRAM    LISTEN      8952   /dev/log
unix  2       [ ]        DGRAM    LISTEN      9047   /var/run/dbus/system_
unix  3       [ ]        STREAM   CONNECTED   19773   /var/run/dbus/system_
bus_socket
unix  3       [ ]        STREAM   CONNECTED   9046   /var/run/dbus/system_
bus_socket
unix  3       [ ]        STREAM   CONNECTED   9487   /var/run/dbus/system_
bus_socket
unix  3       [ ]        STREAM   CONNECTED   9486   /var/run/dbus/system_
unix  3       [ ]        STREAM   CONNECTED   9045   /var/run/dbus/system_
unix  3       [ ]        STREAM   CONNECTED   19533   /var/run/dbus/system_
unix  3       [ ]        STREAM   CONNECTED   9582   /var/run/dbus/system_
unix  3       [ ]        STREAM   CONNECTED   9039   /var/run/dbus/system_
bus_socket
unix  3       [ ]        STREAM   CONNECTED   19598   /var/run/dbus/system_
bus_socket
unix  3       [ ]        STREAM   CONNECTED   9503   /var/run/dbus/system_
bus_socket
unix  3       [ ]        STREAM   CONNECTED   9038

```

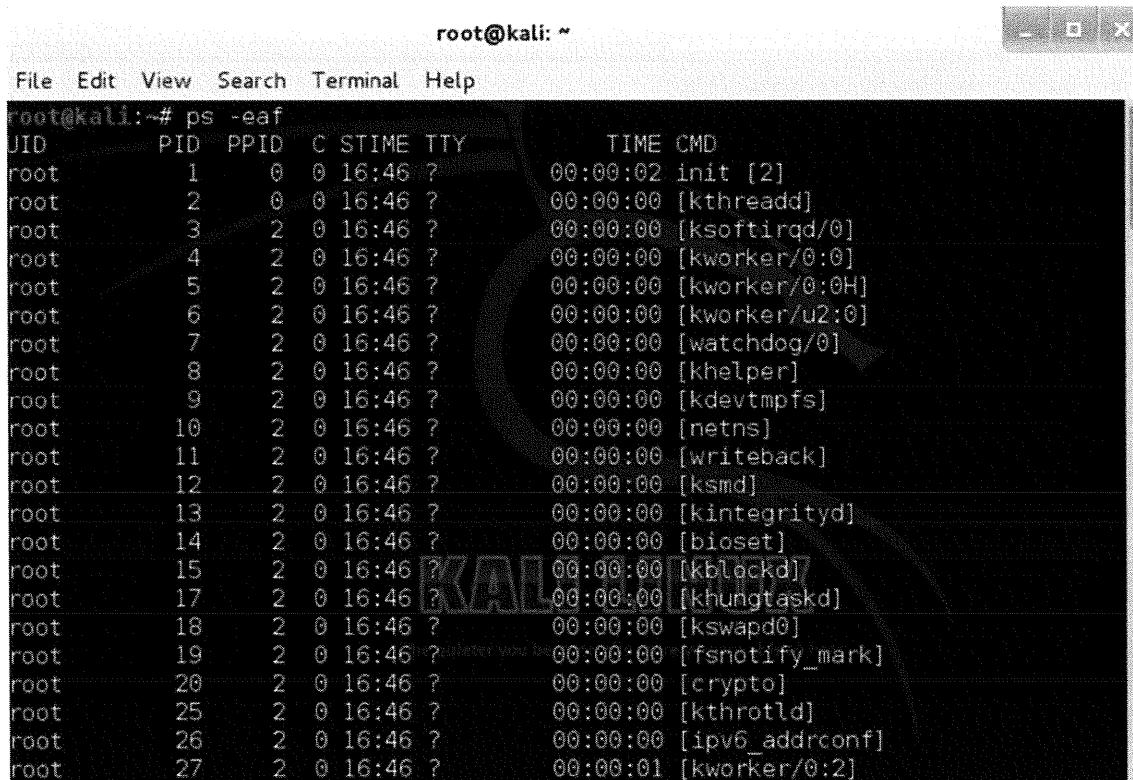
The **-an** option that was added to netstat specifies to list all connections (**-a**) and does not try to resolve hostnames (**-n**).

Depending on the applications you are running, the output might not fit onto one screen. This is a great time to pipe the output of one command through another. You can issue the netstat **-an | more** command to show one screen of information at a time, or you can use grep to search the output for specific requirements. The netstat **-an | grep LISTEN** command outputs all of the servers that are listening on your system, as shown in the following screen. Besides LISTEN, some of the other possible states are ESTABLISHED and TIME_WAIT.

```
root@kali:~# netstat -an | grep LISTEN
unix 2 [ ACC ] STREAM LISTENING 19577 @/tmp/dbus-bEb4POUHcn
unix 2 [ ACC ] STREAM LISTENING 9199 /tmp/.X11-unix/X0
unix 2 [ ACC ] STREAM LISTENING 19588 @/tmp/.ICE-unix/4025
unix 2 [ ACC ] STREAM LISTENING 9198 @/tmp/.X11-unix/X0
unix 2 [ ACC ] STREAM LISTENING 9017 /var/run/dbus/system_
bus_socket
unix 2 [ ACC ] STREAM LISTENING 19563 /tmp/ssh-YsTRyV4kmTLx
/agent.4025
unix 2 [ ACC ] SEQPACKET LISTENING 4458 /run/udev/control
comm
unix 2 [ ACC ] STREAM LISTENING 9350 /var/run/pcscd/pcscd.
blueF
unix 2 [ ACC ] STREAM LISTENING 9467 @/tmp/gdm-session-IxR
S7/native
unix 2 [ ACC ] STREAM LISTENING 19589 /tmp/.ICE-unix/4025
unix 2 [ ACC ] STREAM LISTENING 20494 /tmp/pulse-QdxEC1bd00
S7/dbus-socket
unix 2 [ ACC ] STREAM LISTENING 20499 /tmp/pulse-QdxEC1bd00
unix 2 [ ACC ] STREAM LISTENING 19526 @/tmp/dbus-uJWm9vH08z
unix 2 [ ACC ] STREAM LISTENING 19307 /root/.cache/keyring-
Do3JmE/control
unix 2 [ ACC ] STREAM LISTENING 19674 /root/.cache/keyring-
Do3JmE/gpg
```

The ps Command

Linux gives you the ability to run a command in the background by adding a blank space, and then the & symbol to the end of the command. To obtain a listing of currently running processes, including those that are running in the background, Linux provides the ps command. This command is invaluable for troubleshooting and for determining the current state of the system. Many options can be given to ps to control what it outputs to the command shell. For example, type **ps -ef** into a command shell and press *Enter*, as in the following screen.



A terminal window titled "root@kali: ~" showing the output of the command "ps -eaf". The window has a standard Linux terminal interface with a menu bar at the top.

JID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	16:45	?	00:00:02	init [2]
root	2	0	0	16:46	?	00:00:00	[kthreadd]
root	3	2	0	16:46	?	00:00:00	[ksoftirqd/0]
root	4	2	0	16:46	?	00:00:00	[kworker/0:0]
root	5	2	0	16:46	?	00:00:00	[kworker/0:0H]
root	6	2	0	16:46	?	00:00:00	[kworker/u2:0]
root	7	2	0	16:46	?	00:00:00	[watchdog/0]
root	8	2	0	16:46	?	00:00:00	[khelper]
root	9	2	0	16:46	?	00:00:00	[kdevtmpfs]
root	10	2	0	16:46	?	00:00:00	[netns]
root	11	2	0	16:46	?	00:00:00	[writeback]
root	12	2	0	16:46	?	00:00:00	[ksmd]
root	13	2	0	16:46	?	00:00:00	[kintegrityd]
root	14	2	0	16:46	?	00:00:00	[bioset]
root	15	2	0	16:46	?	00:00:00	[kblockd]
root	17	2	0	16:46	?	00:00:00	[khungtaskd]
root	18	2	0	16:46	?	00:00:00	[kswapd0]
root	19	2	0	16:46	?	00:00:00	[fsnotify_mark]
root	20	2	0	16:46	?	00:00:00	[crypto]
root	25	2	0	16:46	?	00:00:00	[Kthrotld]
root	26	2	0	16:46	?	00:00:00	[ipv6_addrconf]
root	27	2	0	16:46	?	00:00:01	[kworker/0:2]

Because there is a lot of information, which command would you type to determine whether any crypto process is running?

If you said, **ps -eaf | grep crypto**, you are correct.

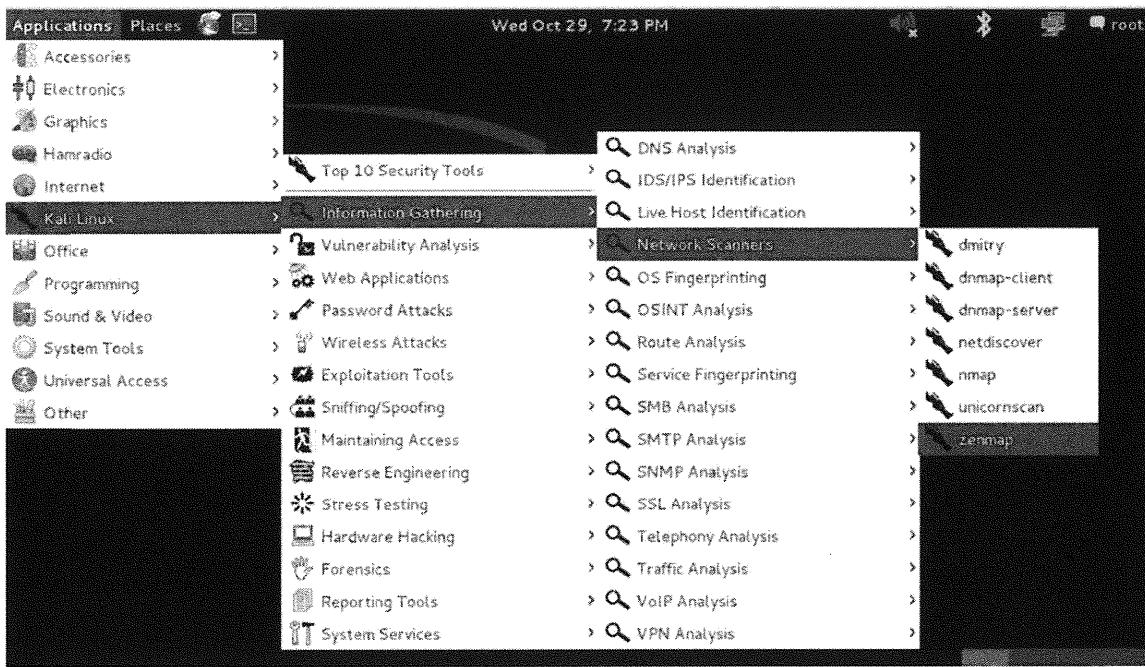


A terminal window titled "root@kali: ~" showing the output of the command "ps -ef | grep crypto". The window has a standard Linux terminal interface with a menu bar at the top.

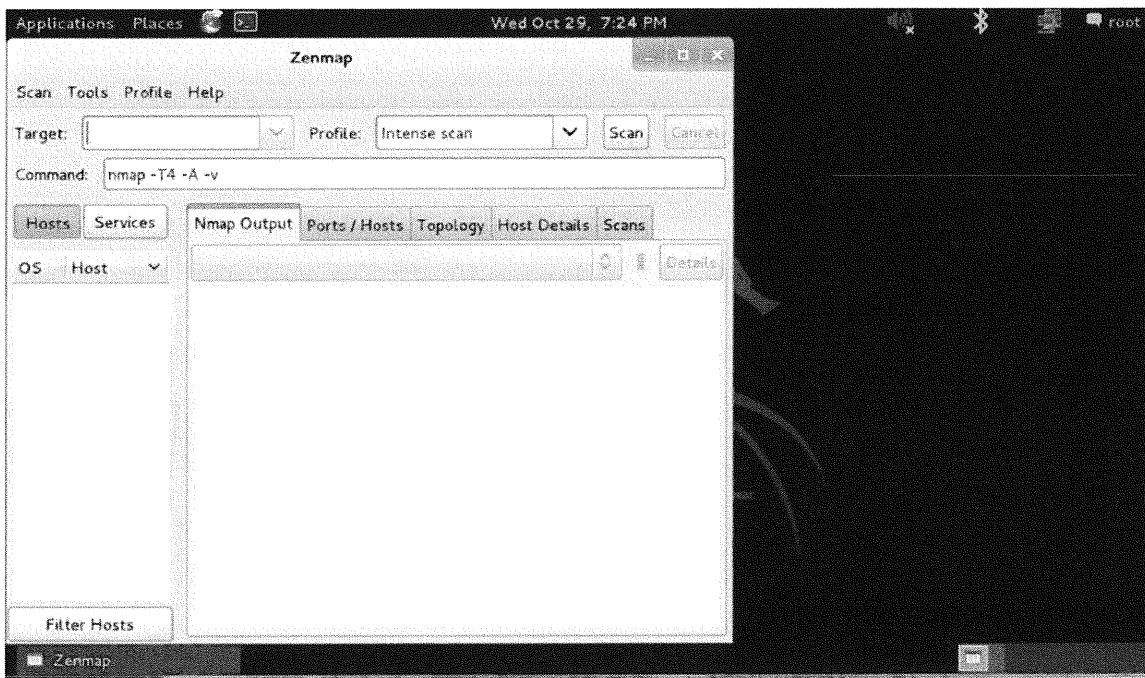
File	Edit	View	Search	Terminal	Help
root@kali:~# ps -ef grep crypto					
root	20	2	0	16:46	?
root	4587	4541	0	19:22	pts/0
					00:00:00 [crypto]
					00:00:00 grep crypto

If a program becomes unresponsive, you can forcibly end it. To do so, you first need to know what process id (PID) it is using. To determine this, issue the **ps -eaf** command and find the entry for the unresponsive program. The second column from the left contains the PID. Issue the **kill** command to kill the program.

To put this into practice, let's start a program and identify the PID. From the Applications menu, click *Kali Linux -> Information Gathering -> Network Scanners ->zenmap*.



The zenmap GUI starts.



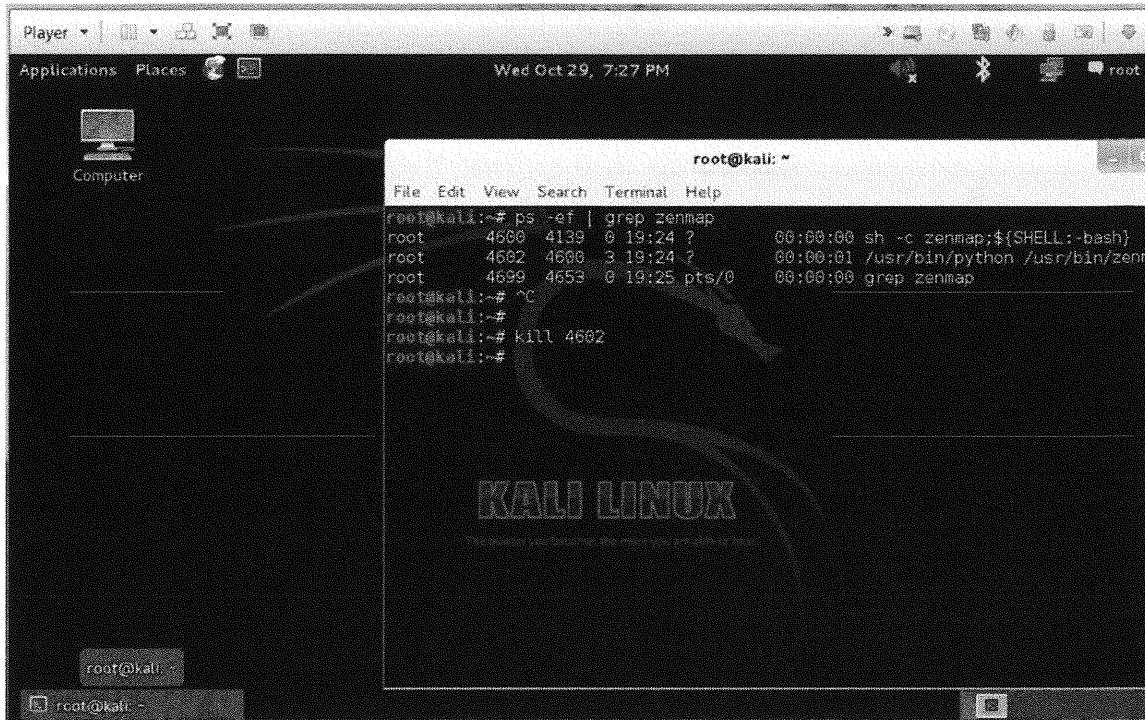
Open a terminal window. From the terminal window, type **ps -ef | grepzenmap** to get a listing of the processes.

```
root@kali:~# ps -ef | grep zenmap
root    4600  4139  0 19:24 ?        00:00:00 sh -c zenmap;${SHELL:-bash}
root    4602  4600  3 19:24 ?        00:00:01 /usr/bin/python /usr/bin/zenmap
root    4699  4653  0 19:25 pts/0    00:00:00 grep zenmap
root@kali:~#
```

The first number listed for the zenmap program is the process ID. In this case, it is 4602, but it might be different on your system. To terminate the process, type **kill 4602** (replacing 4602 with the number of the process on your system).

```
root@kali:~# ps -ef | grep zenmap
root    4600  4139  0 19:24 ?        00:00:00 sh -c zenmap;${SHELL:-bash}
root    4602  4600  3 19:24 ?        00:00:01 /usr/bin/python /usr/bin/zenmap
root    4699  4653  0 19:25 pts/0    00:00:00 grep zenmap
root@kali:~# ^C
root@kali:~#
root@kali:~# kill 4602
```

Press *Enter* to close the zenmap window.



You now have a basic knowledge of how processes work.

You should also now have a basic understanding of Linux. As is the case with anything in life, the best way to understand a topic is to practice. The information contained in this section provides you with a basic knowledge to navigate the file system, perform basic configuration changes, and install applications. There are many security tools written for Linux. Taking the time to learn the tools provides a powerful (yet free) toolbox for assessing the security of your network.

tcpdump

tcpdump is a sniffer that comes with most versions of Linux and allows you to capture packets and do basic filtering.

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

tcpdump

What traffic is flowing across your network? This is a question many network administrators cannot answer. As a result, many administrators do not have the ability to determine what is happening in the event of an attack; and even worse, they cannot be alerted unless the attacker causes damage. There are many organizations that capture network traffic and analyze it to determine detailed information about a previous compromised system. No matter how you use the information, such network traffic information can provide you with vital information and vital clues.

Tcpdump is a command-line tool that can monitor and capture network traffic. Its versatility includes the ability to capture either all traffic that passes over the network segment on which it resides, or only traffic that meets certain criteria. Tcpdump has the ability, through libpcap (a packet capture driver), to capture general traffic by forcing your NIC into promiscuous mode. Or, you can keep your NIC in normal mode and capture only traffic that is destined to and originating from your machine.

Details of tcpdump

- Name: tcpdump
- Operating system: Linux Windows (windump)
- License: freeware
- Protocols used: IP, TCP, UDP, and ICMP
- Category: sniffer
- Description: tcpdump is a sniffer that allows you to analyze most layer 3 and layer 4 protocols. It also has built in filtering capabilities
- URLs: <http://www.tcpdump.org> and <http://www.winpcap.org/windump/>

SANS Security Essentials - © 2010 Secure Anchor Consulting LLC

Details of tcpdump

The following topics and action items are covered in this chapter:

- Understand tcpdump and its many uses.
- Identify the common options used in tcpdump.
- Work through examples of tcpdump running.
- Practice running tcpdump using differing options.

Background of tcpdump

- tcpdump is a command-line tool
- It will sniff all traffic it can see, so a switch could cause problems
- It allows for troubleshooting problems and isolating bottlenecks
- It allows for filtering out certain traffic on high bandwidth segments

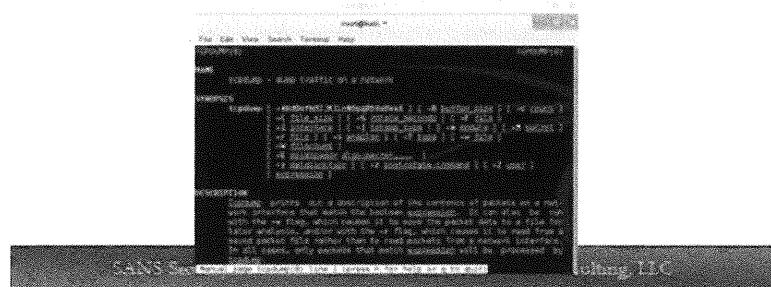
SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

Background of tcpdump

This section intentionally left blank.

Purpose of tcpdump

- It allows you to analyze traffic on a network
- Remember that tcpdump does not show you all of the header fields (by default)



A screenshot of a terminal window titled "Terminal" showing network traffic analysis. The window displays several lines of raw network data, likely in hex and ASCII format. Below the terminal window, a banner reads "SANS Security Institute" and "SANS, LLC".

Purpose of tcpdump

Tcpdump is an invaluable tool when determining the normal behavior of a system or a network. Unlike an Intrusion Detection System (IDS), tcpdump and sniffers in general collect only the network data; it is up to the administrator to then analyze that information and compare his or her findings to baselines that have been collected.

Architecture of tcpdump

- tcpdump sits on top of libpcap or winpcap (for Windows)
- Libpcap actually pulls the bits off of the wire, and tcpdump processes them
- You must make sure that the low level driver is installed and bound to your NIC card in order for it to work properly

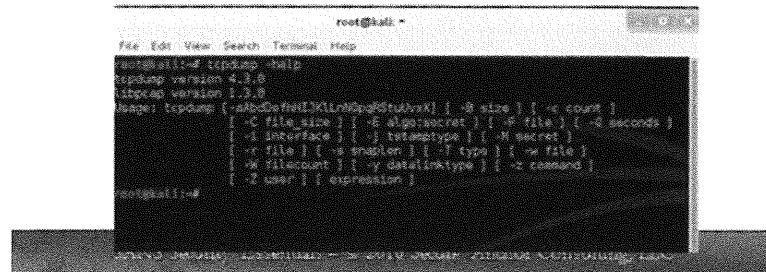
SANS Security Essentials – © 2010 Secure Anchor Consulting, LLC

Architecture of tcpdump

This section intentionally left blank.

Installation

- Installed by default with Kali
- Run tcpdump from the command line:



A terminal window titled "root@kali:" showing the help output for the tcpdump command. The output includes the version information (tcpdump version 4.3.0, libpcap version 1.3.8), usage examples, and various options for capturing network traffic.

```
root@kali: ~
File Edit View Search Terminal Help
tcpdump --help
tcpdump version 4.3.0
libpcap version 1.3.8
Usage: tcpdump [-abBChHILNnRStUvVxX] [-B size] [-c count]
               [-C file_size] [-D signature] [-E file] [-G seconds]
               [-I interface] [-j timestamp] [-M magic]
               [-n file] [-r snapshot] [-T type] [-w file]
               [-X filecount] [-Y datalinktype] [-z command]
               [-Z user] [-e expression]

tcpdump(1) ->
```

Installation

Tcpdump is automatically installed with Kali. Its ease of configuration is one of the reasons so many people use tcpdump instead of moving to one of the many GUI packet sniffers. You can use an unlimited number of possible configurations with tcpdump, making it a powerful tool for a network administrator's arsenal. A Windows equivalent is Windump, which can be downloaded from <http://www.winpcap.org/windump/>.

Running tcpdump

- tcpdump has many options
- Use man tcpdump
- Common options:
 - i to specify an interface (usually not needed)
 - c to collect specific number of packets
 - n to not to name resolution
 - v for verbose mode
 - w to write to the output to a file
 - r to read packets from a file
 - s to specify the number of bytes to capture
 - X to display both hex and ASCII
 - H to filter on a specific host
 - N to negate a filtering option

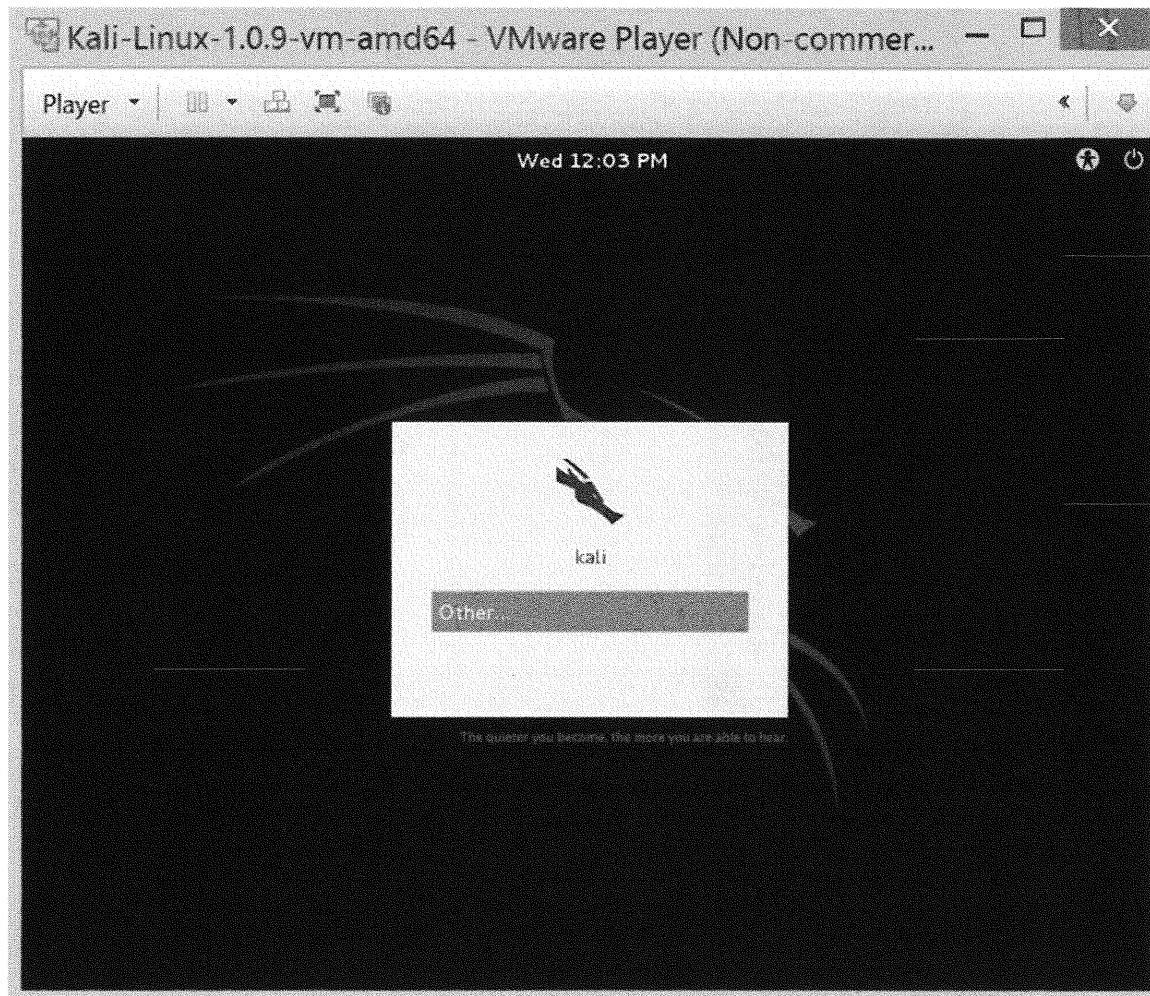
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Running Tcpdump

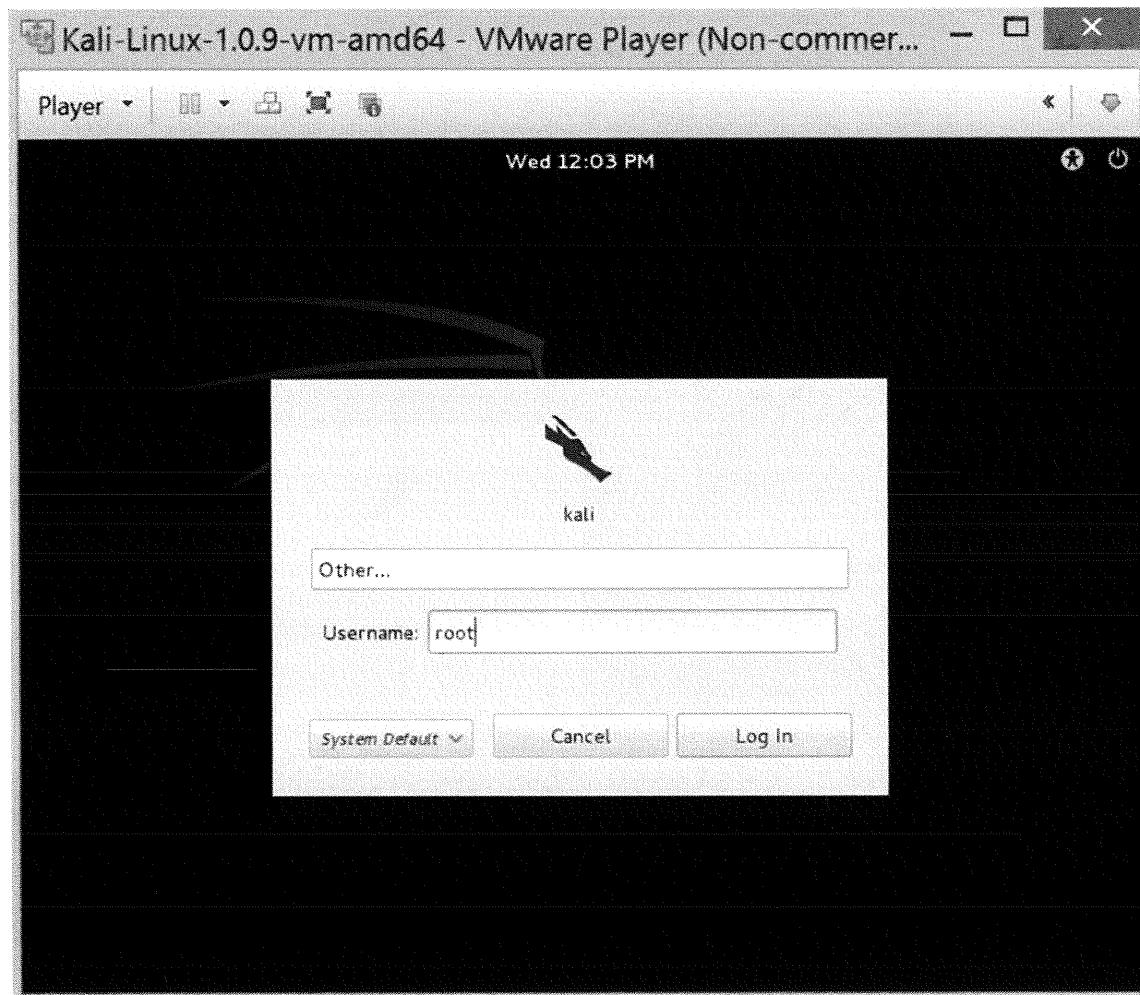
This section explains how to run various command options in tcpdump. The discussion starts with a lesson on how to log in as root, and then shows you how to find more information using the man pages. You also learn how to capture traffic and how to use each of the options listed in the previous illustration.

To open a root shell and run tcpdump, perform the following steps:

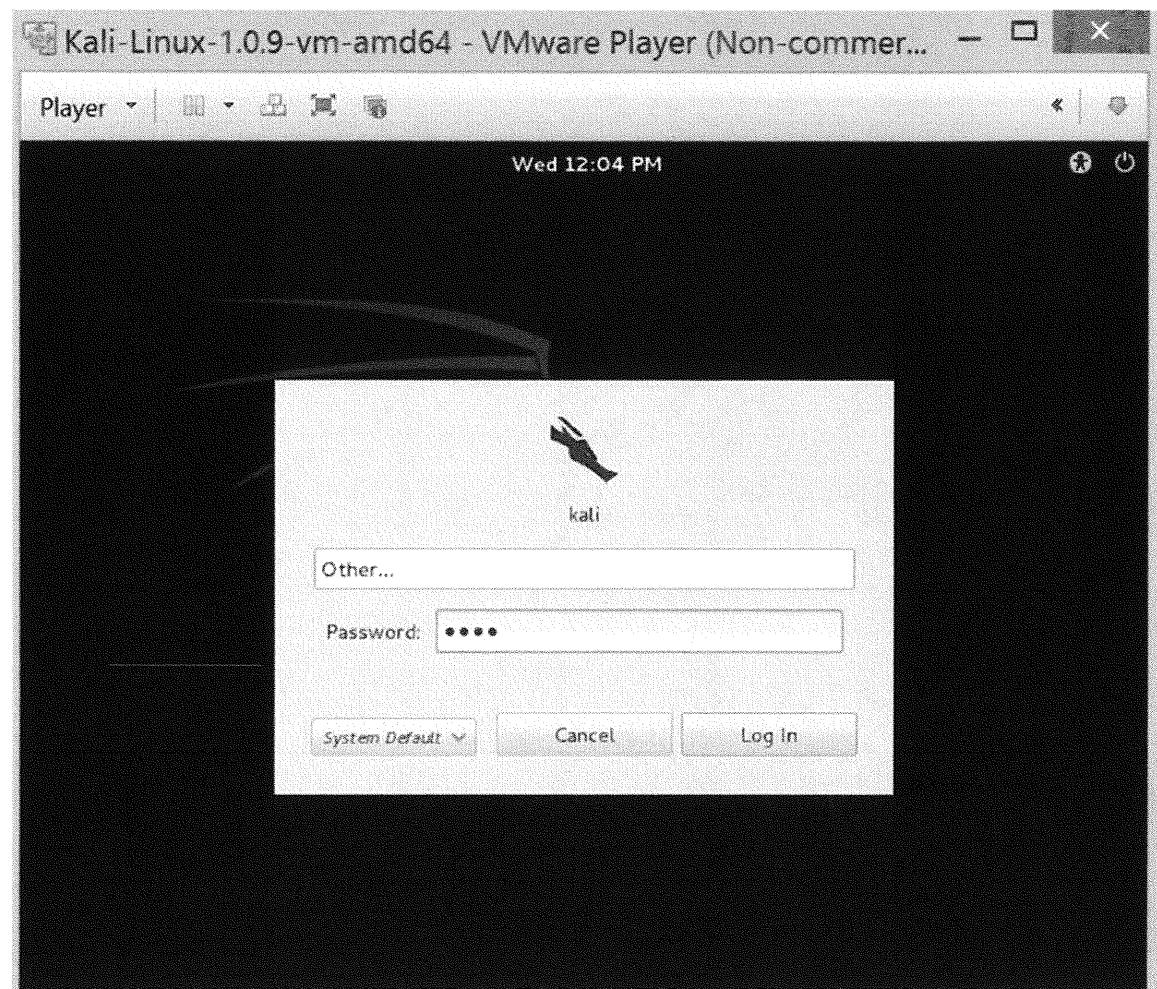
1. After the system boots, you are required to log on to the system. The following shows the specific steps for doing this.



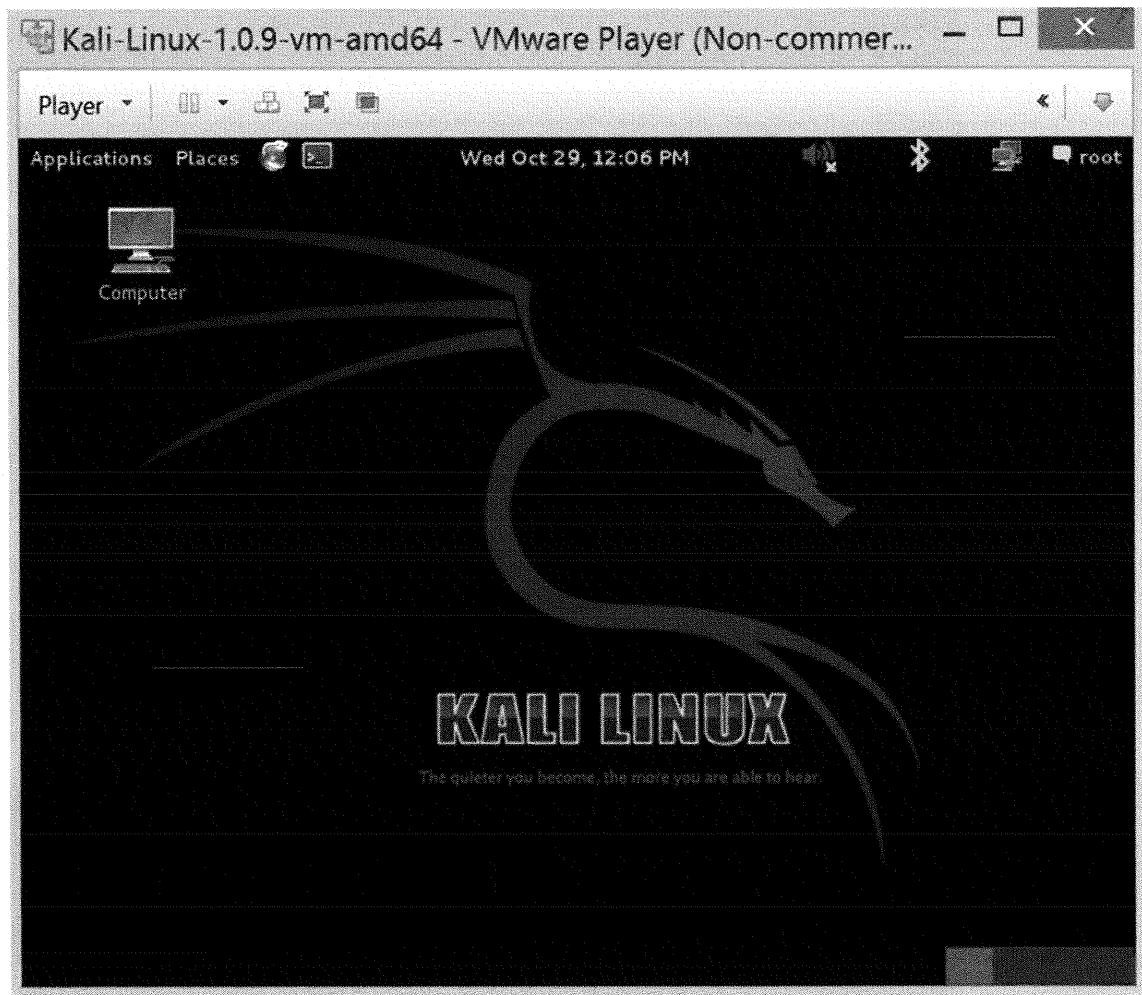
2. After Kali starts, click *Other*, type **root** as the username, and click *Log In*.



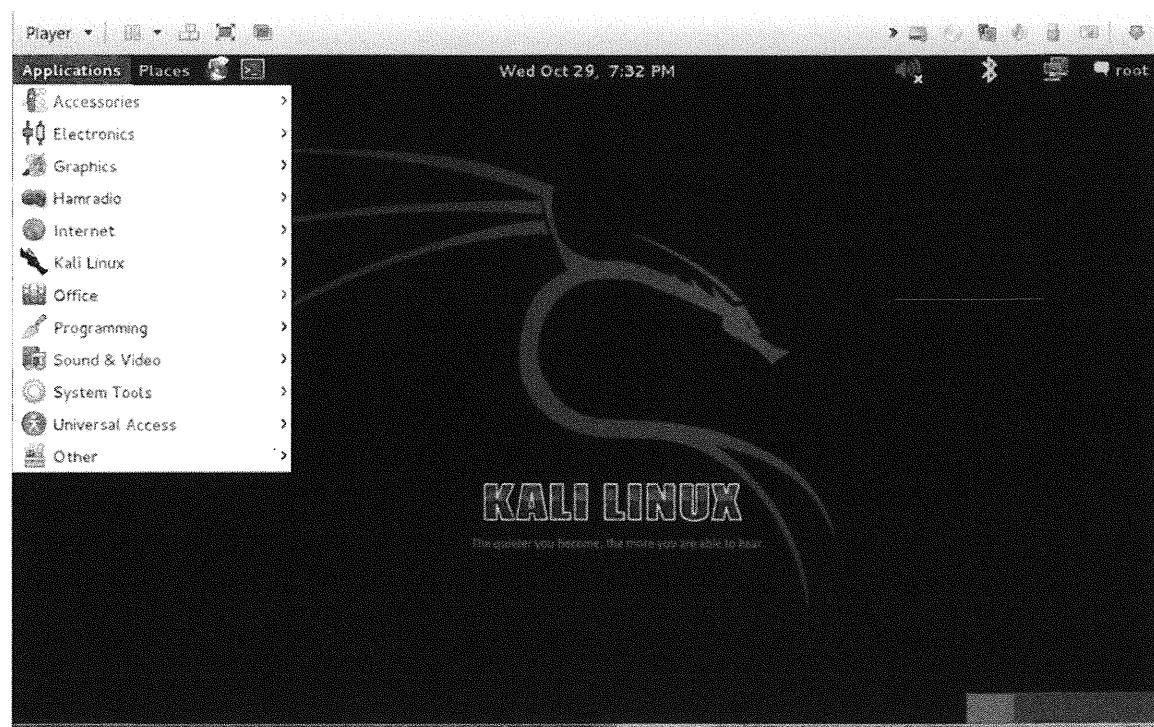
-
3. When prompted for a password, type in **toor** (which is root backwards), and click *Log In*.



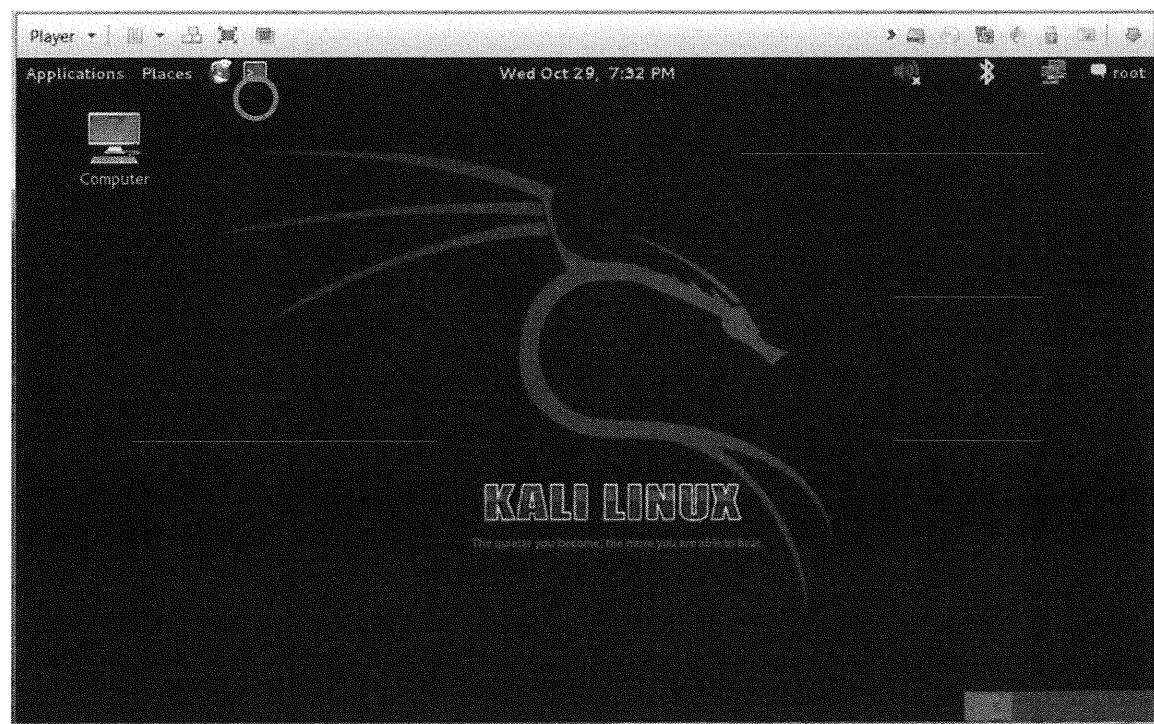
4. Kali Linux starts and you are ready to run the lab.



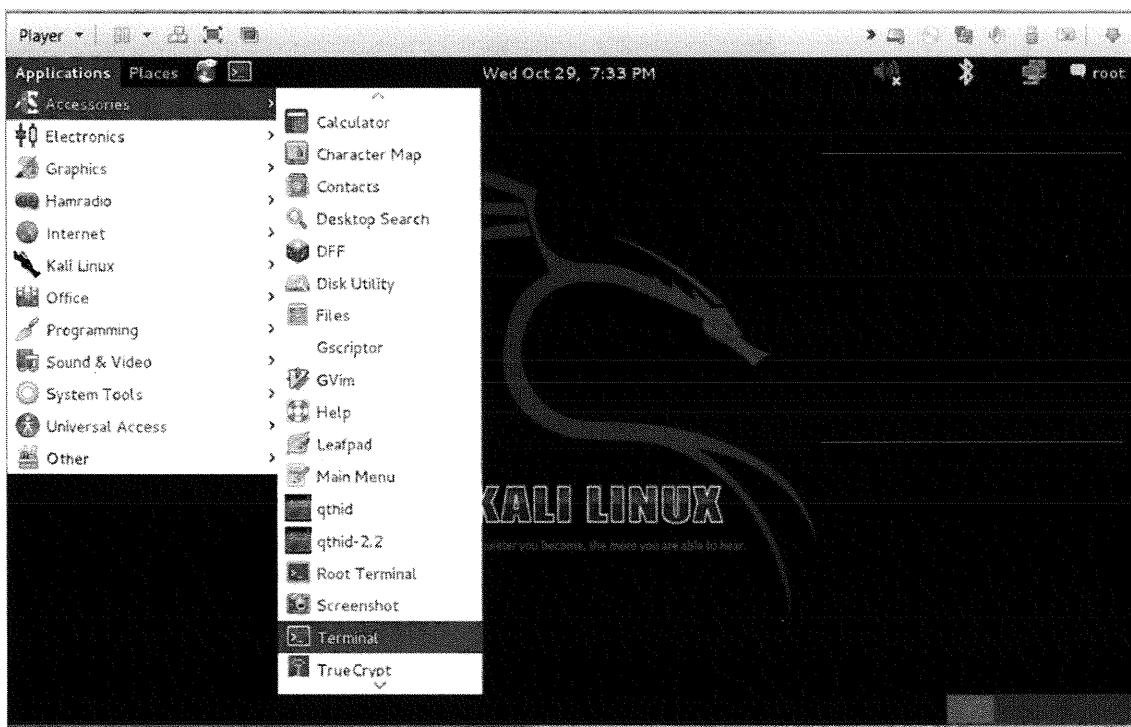
-
5. Now that you are in the graphical interface, click the *Applications* menu to select which applications you would like to run.



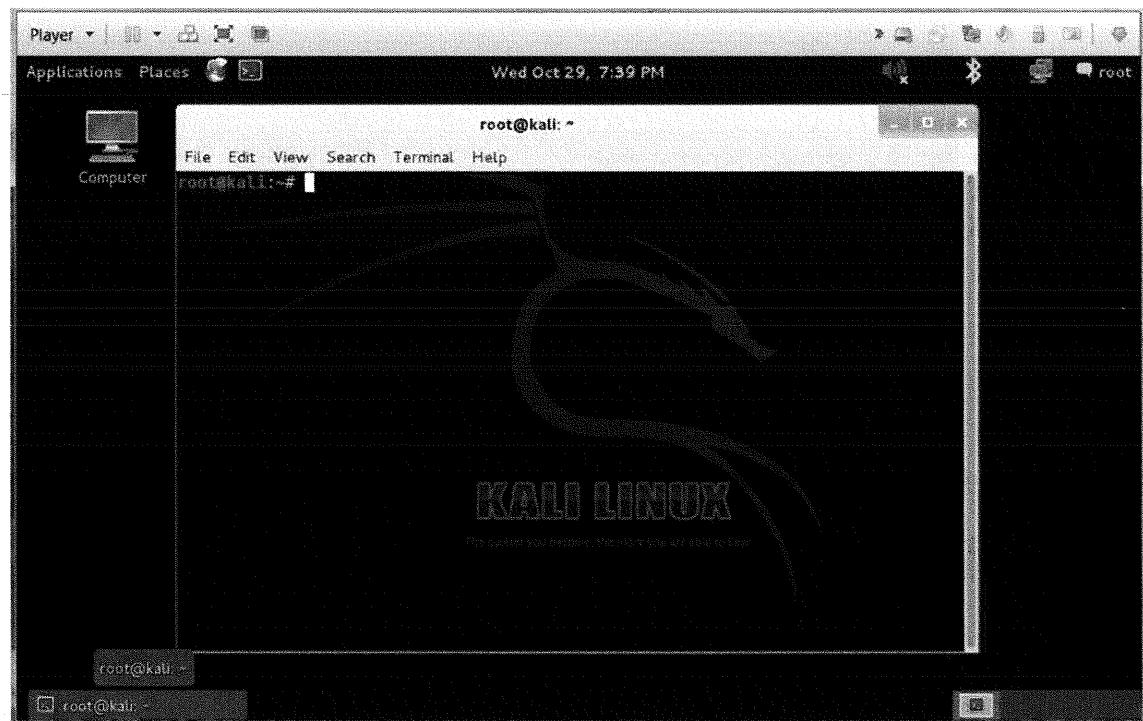
6. To start a sniffer shell or a terminal window, after the word “System” on the Kali menu, click the icon.



7. For another option, click the *Applications* menu, select *Accessories*, and select *Terminal*.

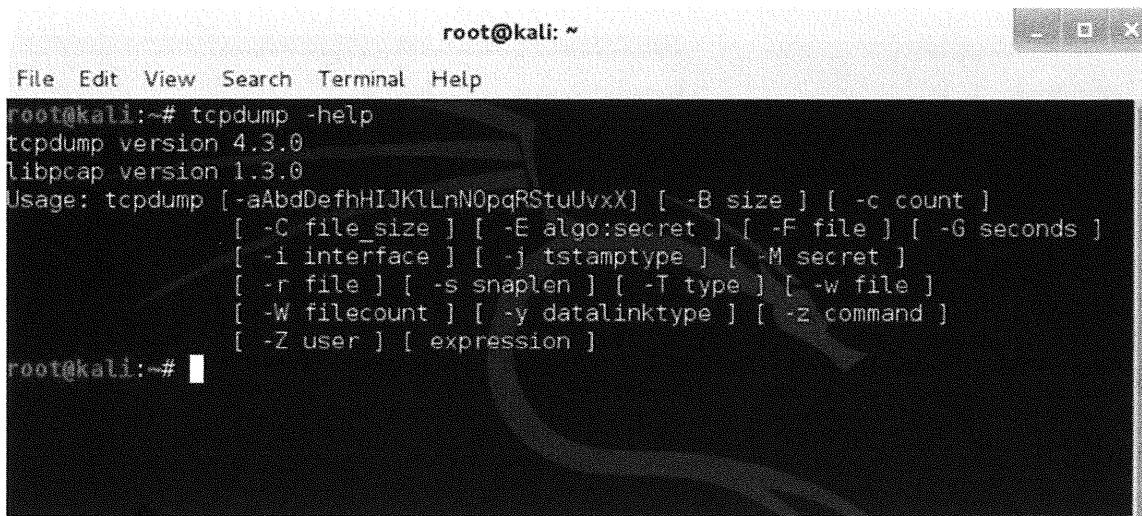


8. After a terminal window opens, it displays in Kali.



9. Now that you have a terminal window open and you are logged in as root, you can play around with tcpdump. To see all of the possible flags or options associated with tcpdump, type the following:

tcpdump -help



A screenshot of a terminal window titled "root@kali:~". The window shows the command "tcpdump -help" being run, followed by the usage information for the tool. The usage text includes various flags like -a, -B, -c, -C, -E, -F, -G, -i, -j, -M, -r, -s, -T, -w, -W, -y, and -Z, along with descriptions of their functions.

```
root@kali:~# tcpdump -help
tcpdump version 4.3.0
libpcap version 1.3.0
Usage: tcpdump [-aAbdDefhHIJKLMNOPpqRStuUvxX] [ -B size ] [ -c count ]
           [ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
           [ -i interface ] [ -j timestamptype ] [ -M secret ]
           [ -r file ] [ -s snaplen ] [ -T type ] [ -w file ]
           [ -W filecount ] [ -y datalinktype ] [ -z command ]
           [ -Z user ] [ expression ]
root@kali:~#
```

10. To view the man page for tcpdump, type the following:

man tcpdump

While viewing the following man (manual) page, there are a few different ways you can move around. If you press *Enter*, you will progress through the document one line at a time. If you press the *spacebar*, you will progress one page at a time. You can exit the man page at any time by typing **q**. Spend a few minutes reviewing the different types of information on the man page. Using the man pages are also a helpful way to find information on a tool if you ever get stuck and need more details on how to use the tool.

```
root@kali: ~
File Edit View Search Terminal Help
TCPDUMP(8)                                     TCPDUMP(8)
NAME
    tcpdump - dump traffic on a network

SYNOPSIS
    tcpdump [ -AbdDefHJKILLnNOpqRStuvvxX ] [ -B buffer size ] [ -c count ]
    [ -C file size ] [ -G rotate seconds ] [ -F file ]
    [ -i interface ] [ -j tstamp type ] [ -m module ] [ -M secret ]
    [ -r file ] [ -s snapshot ] [ -T type ] [ -w file ]
    [ -W filecount ]
    [ -E spi@ipaddr algo:secret... ]
    [ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]
    [ expression ]

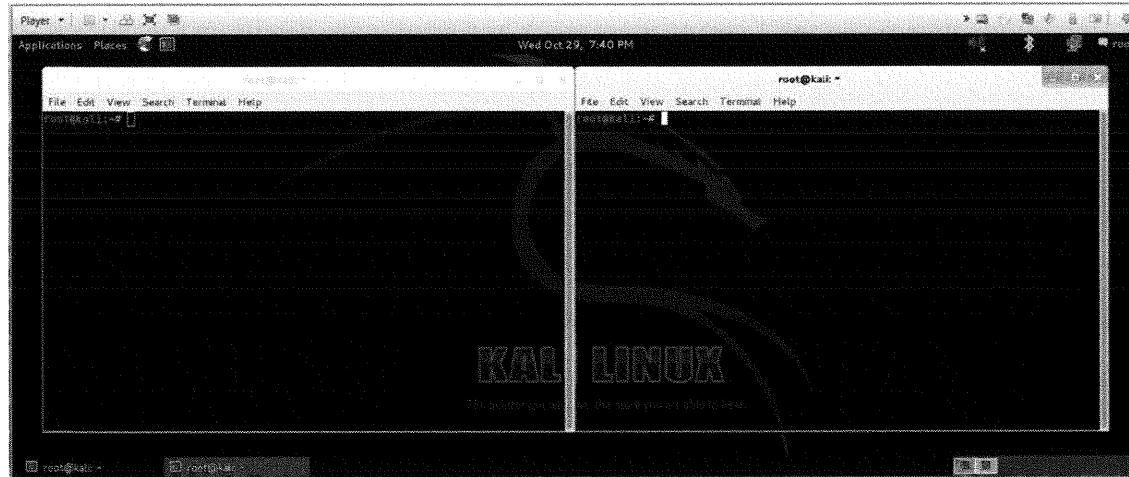
DESCRIPTION
    Tcpdump prints out a description of the contents of packets on a net-
    work interface that match the boolean expression. It can also be run
    with the -w flag, which causes it to save the packet data to a file for
    later analysis, and/or with the -r flag, which causes it to read from a
    saved packet file rather than to read packets from a network interface.
    In all cases, only packets that match expression will be processed by
    tcpdump.

Manual page tcpdump(8) line 1 (press h for help or q to quit)
```

As you can see, there are many options associated with `tcpdump`. This chapter does not cover every one of them, but you do learn about the more commonly used flags or options.

Capturing Traffic

Open two terminal windows as described previously. For example, click the *Applications* menu, click *Accessories*, and click *terminal*—perform this twice to open up two windows. Use one shell to run `tcpdump` and the other shell to generate traffic.



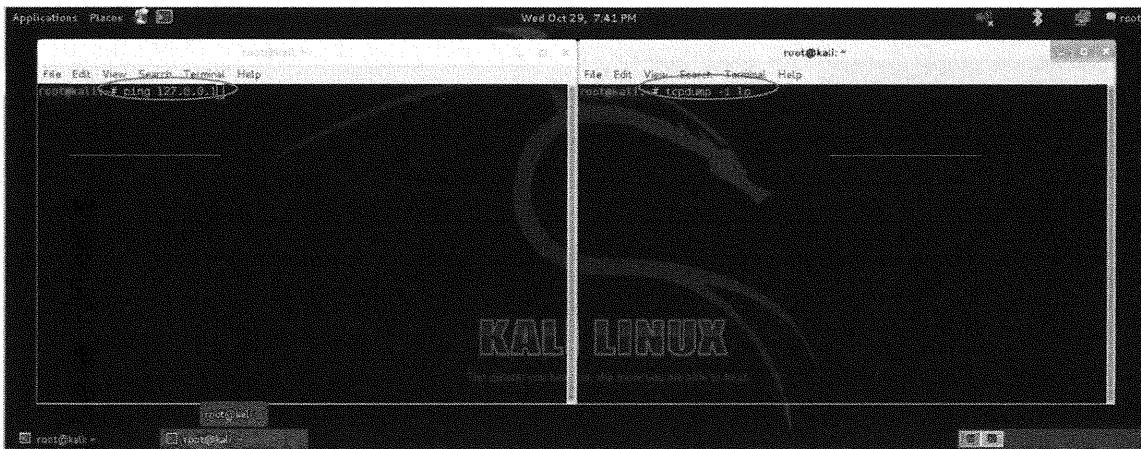
In this exercise, you use `tcpdump` to capture traffic by sending traffic to the localhost or loopback address (127.0.0.1).

1. To keep things simple, you ping your loopback address, so you do not need an external address.
2. In the first window (on the left), type the following:

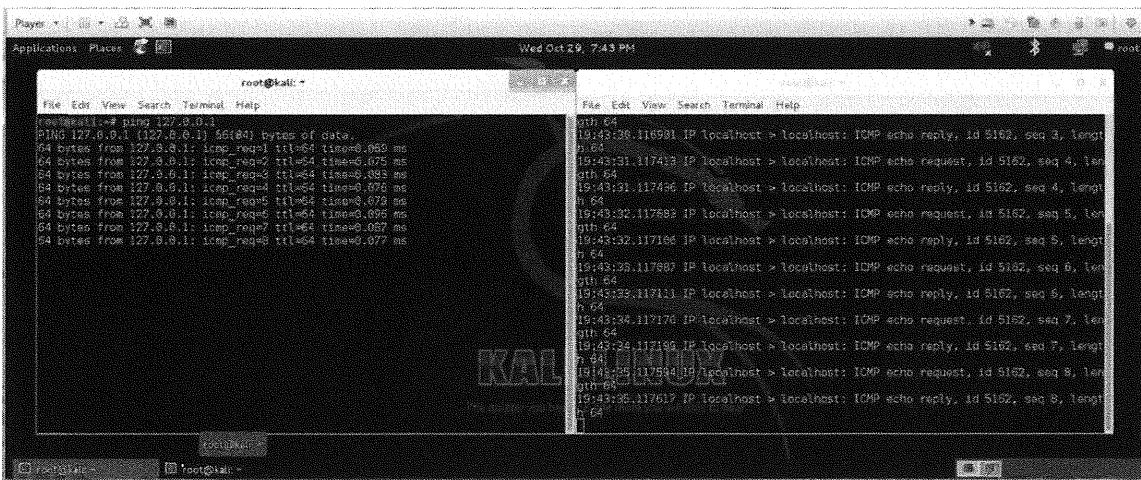
ping 127.0.0.1

This actively pings your loopback adapter, and it will not stop until you tell it to stop by pressing *CTRL-C* (this means hold down the *CTRL* key and the *C* key at the same time). Next, move your cursor over to the second terminal window (on the right) and type the following:

tcpdump -i lo

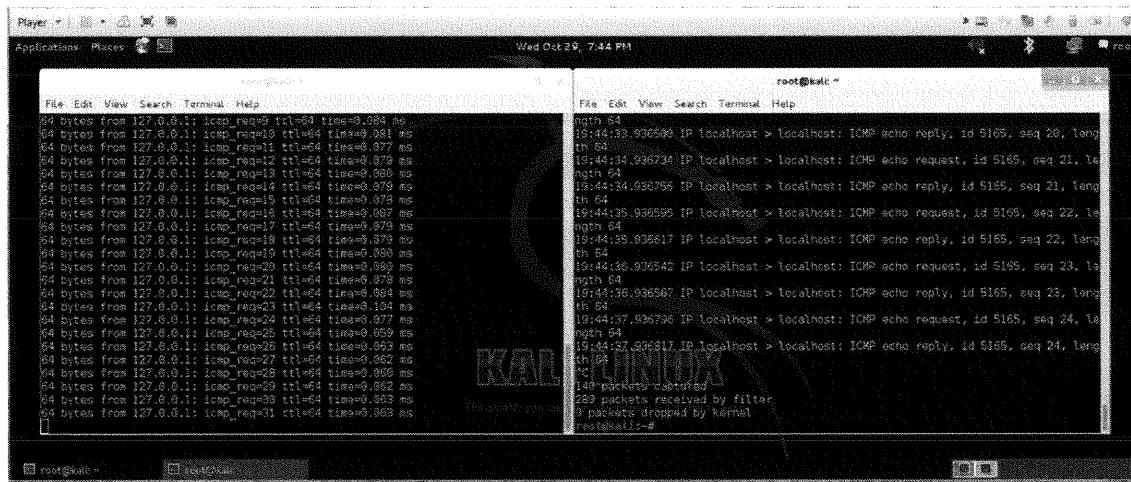


Now, the window on the left is generating ping traffic and the window on the right is showing the tcpdump output from that traffic.



Because tcpdump has several options, you try each of the following flags or options individually so that you can see the respective outputs. Remember that everything in Linux is case sensitive, so type the commands exactly as you see them printed.

For traffic to be generated, keep the left window generating ICMP ping traffic, so that you can focus your work on the right window, which is generating tcpdump commands and sniffing traffic. To stop commands in Linux, press **CTRL-C**. To type additional tcpdump commands, go to the right window in which tcpdump is running and type **CTRL-C**. You should be returned to a command prompt. Remember to keep the left side window pinging so you can generate traffic.



Tcpdump Options

The exercises in the following sections show you how to perform common tasks with tcpdump options.

-i Option

For this first exercise, you learn how to specify an interface with the **-i** option.

The **-i** option means "listen on interface." If unspecified, tcpdump searches the system interface list for the lowest numbered and configured up interface (excluding loopback). The search chooses the most recent match.

This **-i** option is helpful if you have multiple NICs in your system. You can specify a particular interface, such as **lo** or **eth1**. By default, **eth0** is used. With Kali, if you do not specify an interface, it will use **eth1** and the labs will not work. The reason is that you are generating traffic on the loopback address (127.0.0.1) not the Ethernet interface.

```
root@kali:~# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
19:48:08.460334 IP 192.168.86.1.63257 > 239.255.255.250.1900: UDP, length 117
19:48:08.462105 IP 192.168.86.128.45583 > 192.168.86.2.domain: 59848+ PTR? 250.2
55.255.239.in-addr.arpa. (46)
19:48:08.466358 ARP, Request who-has 192.168.86.128 tell 192.168.86.2, length 46
19:48:08.466406 ARP, Reply 192.168.86.128 is-at 00:0c:29:c7:c2:3 (oui Unknown),
length 28
19:48:08.466920 IP 192.168.86.2.domain > 192.168.86.128.45583: 59848 NXDomain 0/
0/0 (46)
19:48:08.467795 IP 192.168.86.128.33461 > 192.168.86.2.domain: 2404+ PTR? 1.86.1
68.192.in-addr.arpa. (43)
19:48:08.470486 IP 192.168.86.1.netbios-ns > 192.168.86.255.netbios-ns: NBT UDP
PACKET(137) : QUERY; REQUEST; BROADCAST
19:48:08.470880 IP 192.168.86.2.domain > 192.168.86.128.33461: 2404 NXDomain 0/0
/0 (43)
19:48:08.471747 IP 192.168.86.128.43920 > 192.168.86.2.domain: 40174+ PTR? 2.86.
168.192.in-addr.arpa. (43)
19:48:08.474800 IP 192.168.86.2.domain > 192.168.86.128.43920: 40174 NXDomain 0/
0/0 (43)
19:48:08.475243 IP 192.168.86.128.51541 > 192.168.86.2.domain: 54758+ PTR? 128.8
6.168.192.in-addr.arpa. (45)
19:48:08.478268 IP 192.168.86.2.domain > 192.168.86.128.51541: 54758 NXDomain 0/
```

To try this option, type **tcpdump -i lo**. This command should be done from the right-hand window, making sure that ping is stilling running in the window on the left-hand side.

```
root@kali:~# tcpdump -i lo
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
19:49:10.941170 IP localhost > localhost: ICMP echo request, id 5165, seq 297, length 64
19:49:10.941193 IP localhost > localhost: ICMP echo reply, id 5165, seq 297, length 64
19:49:11.940429 IP localhost > localhost: ICMP echo request, id 5165, seq 298, length 64
19:49:11.940451 IP localhost > localhost: ICMP echo reply, id 5165, seq 298, length 64
19:49:12.940367 IP localhost > localhost: ICMP echo request, id 5165, seq 299, length 64
19:49:12.940389 IP localhost > localhost: ICMP echo reply, id 5165, seq 299, length 64
^C
6 packets captured
12 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

As shown in the previous screen, the device named “localhost” (in the first line) is sending ICMP echo requests to a device named “localhost.” Essentially, your system is sending packets to itself:

14:01:26.119445 IP localhost>localhost: ICMP echo request, id 20743, seq 1119, length 64

In the second line, the receiving device is sending an echo reply back to the originating device.

14:01:26.119480 IP localhost>localhost: ICMP echo reply, id 20743, seq 1119, length 64

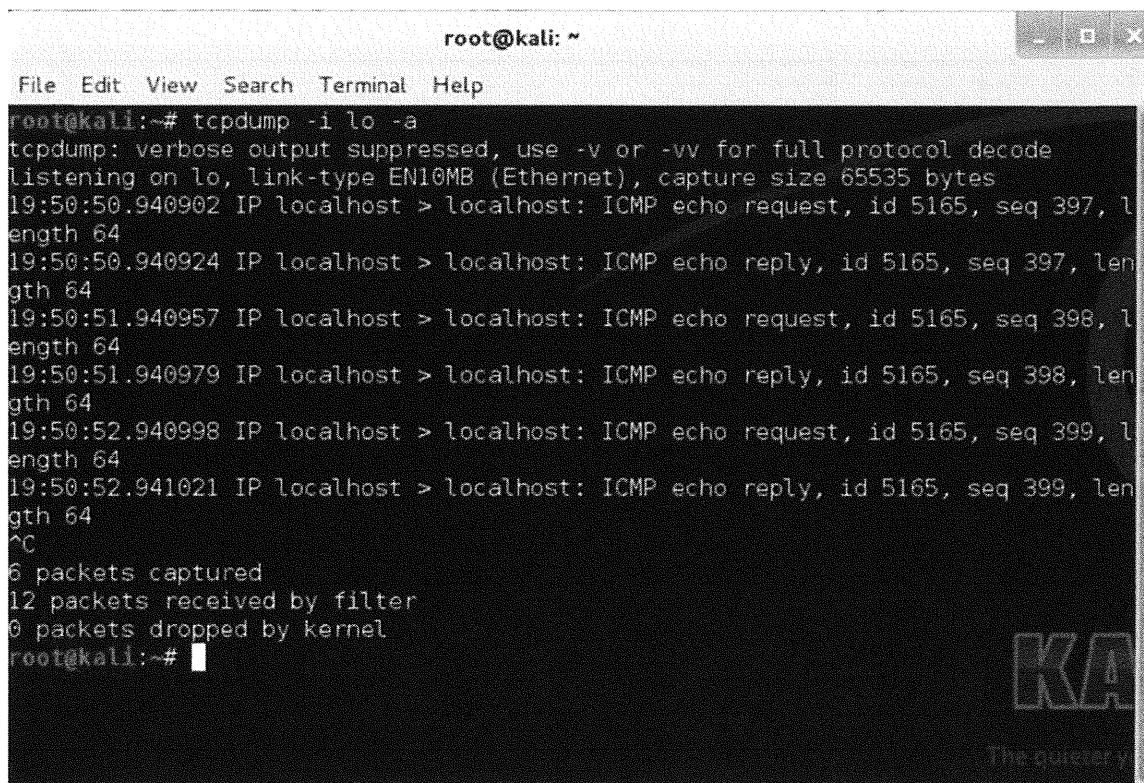
Press **CTRL-C** to stop sniffing traffic with **tcpdump**.

-a Option

The **-a** option attempts to convert network and broadcast addresses to names. Use this flag if you need the proper names rather than IP addresses. However, if you use this flag, you might have packets appear out of order and slower than real time because looking up that information requires additional system resources. For this reason, you might not want to use it.

To try this option, type the following:

tcpdump -i lo -a



A terminal window titled "root@kali: ~" showing the output of the command "tcpdump -i lo -a". The window includes a menu bar with File, Edit, View, Search, Terminal, and Help. The output shows several ICMP echo requests and replies between the localhost interface (lo) and the kernel. The session is terminated with a Ctrl-C interrupt.

```
root@kali:~# tcpdump -i lo -a
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
19:50:50.940902 IP localhost > localhost: ICMP echo request, id 5165, seq 397, length 64
19:50:50.940924 IP localhost > localhost: ICMP echo reply, id 5165, seq 397, length 64
19:50:51.940957 IP localhost > localhost: ICMP echo request, id 5165, seq 398, length 64
19:50:51.940979 IP localhost > localhost: ICMP echo reply, id 5165, seq 398, length 64
19:50:52.940998 IP localhost > localhost: ICMP echo request, id 5165, seq 399, length 64
19:50:52.941021 IP localhost > localhost: ICMP echo reply, id 5165, seq 399, length 64
^C
6 packets captured
12 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

Remember to press *CTRL-C* to stop the current command and return to the root prompt.

-n Option

Name resolution is helpful, but in troubleshooting a problem sometimes it is useful to utilize IP addresses directly. To have IP addresses displayed and not domain names, you can use the **-n** option. From the command prompt, type the following:

```
tcpdump -i lo -n
```

Notice how the output is different from the previous screen.

```
File Edit View Search Terminal Help
root@kali:~# tcpdump -i lo -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
19:51:38.940536 IP 127.0.0.1 > 127.0.0.1: ICMP echo request, id 5165, seq 445, length 64
19:51:38.940559 IP 127.0.0.1 > 127.0.0.1: ICMP echo reply, id 5165, seq 445, length 64
19:51:39.940504 IP 127.0.0.1 > 127.0.0.1: ICMP echo request, id 5165, seq 446, length 64
19:51:39.940524 IP 127.0.0.1 > 127.0.0.1: ICMP echo reply, id 5165, seq 446, length 64
^C
4 packets captured
8 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

Now the IP addresses are shown instead of the system name. Remember to press *CTRL-C* to stop the current command and return to the root prompt.

-c Option

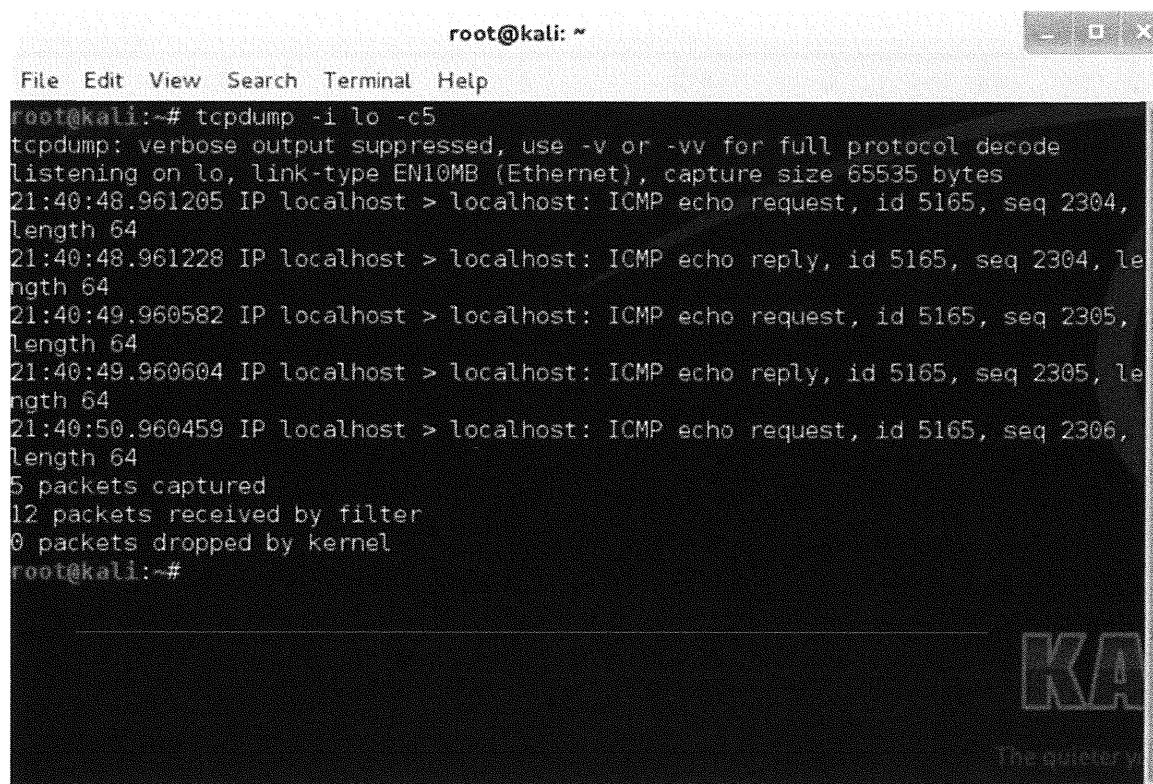
This option means, "Exit after receiving a certain number of packets."

You can use this option to collect a specific number of packets before tcpdump exits. Many administrators use this option in tandem with a batch file to collect specific amounts of network traffic at different times of the day.

To try this option, type the following:

```
tcpdump -i lo -c 5
```

This captures five packets and then stops.



A terminal window titled "root@kali: ~" showing the output of the command "tcpdump -i lo -c5". The output indicates that 5 packets were captured, all being ICMP echo requests and replies between the localhost interface and the loopback interface. The window has a standard Linux terminal interface with a menu bar at the top.

```
root@kali:~# tcpdump -i lo -c5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
21:40:48.961205 IP localhost > localhost: ICMP echo request, id 5165, seq 2304,
length 64
21:40:48.961228 IP localhost > localhost: ICMP echo reply, id 5165, seq 2304, length 64
21:40:49.960582 IP localhost > localhost: ICMP echo request, id 5165, seq 2305, length 64
21:40:49.960604 IP localhost > localhost: ICMP echo reply, id 5165, seq 2305, length 64
21:40:50.960459 IP localhost > localhost: ICMP echo request, id 5165, seq 2306, length 64
5 packets captured
12 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

With the `c` option, you are automatically returned to the root prompt after five packets are captured.

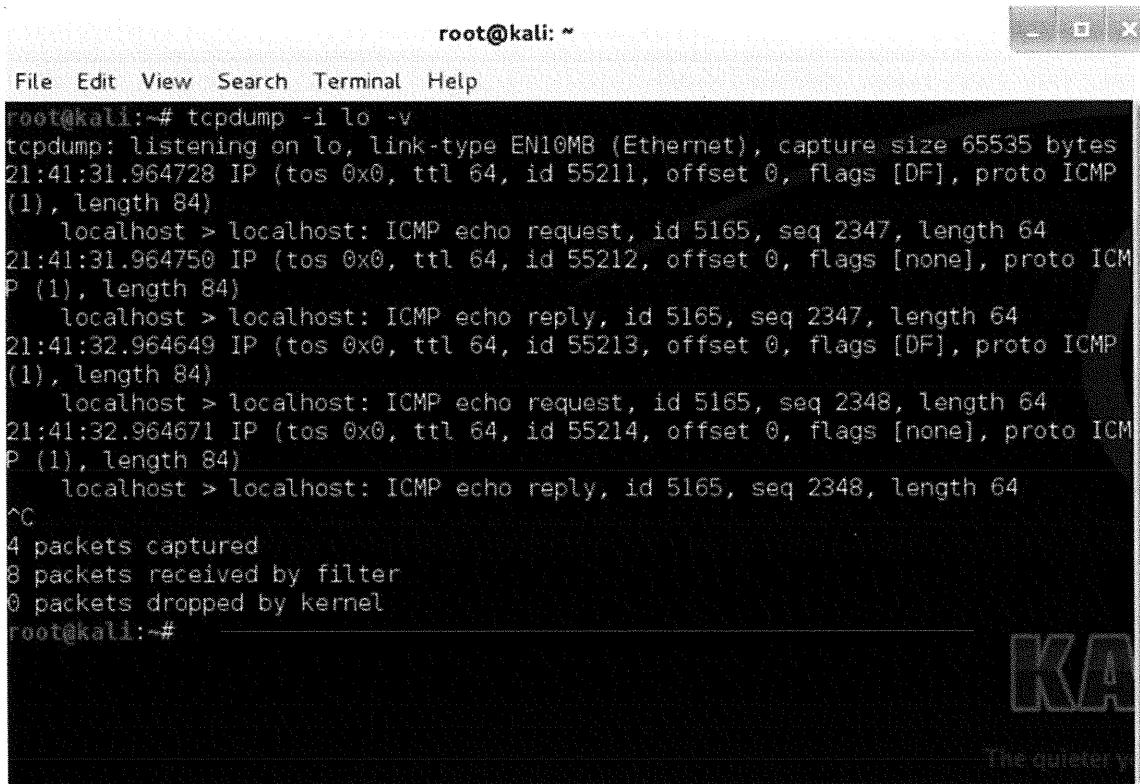
-v Option

This command generates (slightly more) verbose output. For example, it prints the time to live, identification, total length, and options in an IP packet. It also enables additional packet integrity checks, such as verifying the IP and ICMP header checksum.

You can use the `-v` option if you are looking to get as much data as possible from each sniffed packet. It is important to remember that the more information you are capturing and reporting on, the slower the information displays on your screen.

To try this option, type the following:

```
tcpdump -i lo -v
```



```
root@kali:~# tcpdump -i lo -v
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
21:41:31.964728 IP (tos 0x0, ttl 64, id 55211, offset 0, flags [DF], proto ICMP
(1), length 84)
    localhost > localhost: ICMP echo request, id 5165, seq 2347, length 64
21:41:31.964750 IP (tos 0x0, ttl 64, id 55212, offset 0, flags [none], proto ICM
P (1), length 84)
    localhost > localhost: ICMP echo reply, id 5165, seq 2347, length 64
21:41:32.964649 IP (tos 0x0, ttl 64, id 55213, offset 0, flags [DF], proto ICMP
(1), length 84)
    localhost > localhost: ICMP echo request, id 5165, seq 2348, length 64
21:41:32.964671 IP (tos 0x0, ttl 64, id 55214, offset 0, flags [none], proto ICM
P (1), length 84)
    localhost > localhost: ICMP echo reply, id 5165, seq 2348, length 64
^C
4 packets captured
8 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

Notice the additional information that is received, compared to previous outputs. If you are working on an incident, this additional information can be of value.

Remember to press *CTRL-C* to stop the current command and return to the root prompt.

-x Option

The **-x** option prints each packet (minus its link level header) in hex. The smaller of the entire packet, or snaplen bytes, will be printed.

You can use this option if you want the output of `tcpdump` to be in raw hex. This is the fastest option you can choose to get the information in real time; but as you can see, unless you can convert hex into ASCII in your head, the information is not valuable unless you convert every single packet into a readable format.

To try this option, type the following:

tcpdump -i lo -x

```
root@kali:~# tcpdump -i lo -x
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
21:42:18.968614 IP localhost > localhost: ICMP echo request, id 5165, seq 2394,
length 64
    0x0000: 4500 0054 d80a 4000 4001 649c 7f00 0001
    0x0010: 7f00 0001 0800 abc2 142d 095a 7a97 5154
    0x0020: 69c7 0e00 0809 0a0b 0c0d 0e0f 1011 1213
    0x0030: 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
    0x0040: 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
    0x0050: 3435 3637
21:42:18.968637 IP localhost > localhost: ICMP echo reply, id 5165, seq 2394, length 64
    0x0000: 4500 0054 d80b 0000 4001 a49b 7f00 0001
    0x0010: 7f00 0001 0000 b3c2 142d 095a 7a97 5154
    0x0020: 69c7 0e00 0809 0a0b 0c0d 0e0f 1011 1213
    0x0030: 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
    0x0040: 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
    0x0050: 3435 3637
^C
2 packets captured
4 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

Remember to press *CTRL-C* to stop the current command and return to the root prompt.

Now you can practice your decoding skills from class. Remember, we expect byte 0 of an IPv4 packet to be 45, specifying it is IPv4, and it is a 20-byte IP header. Using the TCP/IP reference guide, can you decode the rest of the packet? To practice what we learned in class, if you look at the first packet, hopefully you can decode the entire packet. After validating byte 0, look at the 9th byte. The fact that it has a 1 tells you it is ICMP. Go to where the ICMP header begins, and you can see a type 8 code 0 message, which is an ICMP echo request.

4500 0054 0000 4000 4001 3ca7 7f00 0001

7f00 0001 0800 ffe7 5107 0ac2 e970 524f

738b 0200 0809 0a0b 0c0d 0e0f 1011 1213

1415 1617 1819 1a1b 1c1d 1e1f 2021 2223

2425 2627 2829 2a2b 2c2d 2e2f 3031 3233

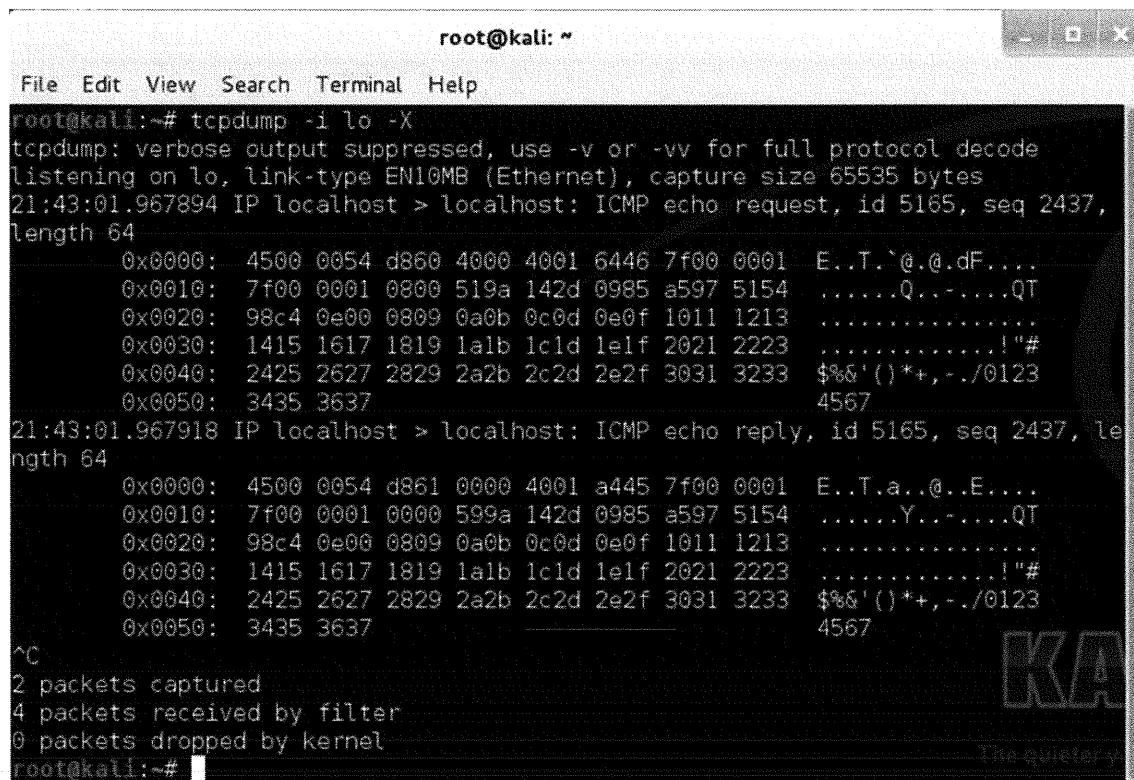
3435 3637

Practice with a few more packets to be comfortable with the decoding.

-X Option

The -X option means that when you print hex, you also print ASCII. You can use this option if you want to see both the hex and the ASCII text at the same time. This causes a lot of information to be put on the screen, so it is generally used only if you write the output to a file. Remember that tcpdump is a sniffer, not a packet decoder like Wireshark. Therefore, you have to manually decode the payload. If you want a little insight into what information is being sent, the -X option displays the ASCII content of the payload. To try this option, type the following:

```
tcpdump -i lo -X
```



A terminal window titled 'root@kali: ~' showing the output of the 'tcpdump -i lo -X' command. The window has a standard Linux terminal interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar. The main area displays network traffic in hex and ASCII format. The output shows two ICMP echo requests and their replies. The first request (id 5165, seq 2437) and its reply (id 5165, seq 2437) are shown in full. The second request (id 5166, seq 2438) and its reply (id 5166, seq 2438) are partially visible. The terminal prompt 'root@kali: ~# ' is at the bottom.

```
root@kali:~# tcpdump -i lo -X
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
21:43:01.967894 IP localhost > localhost: ICMP echo request, id 5165, seq 2437,
length 64
    0x0000: 4500 0054 d860 4000 4001 6446 7f00 0001 E..T.^@.@.dF...
    0x0010: 7f00 0001 0800 519a 142d 0985 a597 5154 .....Q....QT
    0x0020: 98c4 0e00 0809 0a0b 0c0d 0e0f 1011 1213 .....
    0x0030: 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223 ....!#
    0x0040: 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233 $%&(')*+,./0123
    0x0050: 3435 3637                           4567
21:43:01.967918 IP localhost > localhost: ICMP echo reply, id 5165, seq 2437, length 64
    0x0000: 4500 0054 d861 0000 4001 a445 7f00 0001 E..T.a..@..E...
    0x0010: 7f00 0001 0000 599a 142d 0985 a597 5154 .....Y....QT
    0x0020: 98c4 0e00 0809 0a0b 0c0d 0e0f 1011 1213 .....
    0x0030: 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223 ....!#
    0x0040: 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233 $%&(')*+,./0123
    0x0050: 3435 3637                           4567
^C
2 packets captured
4 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

Keyword Options

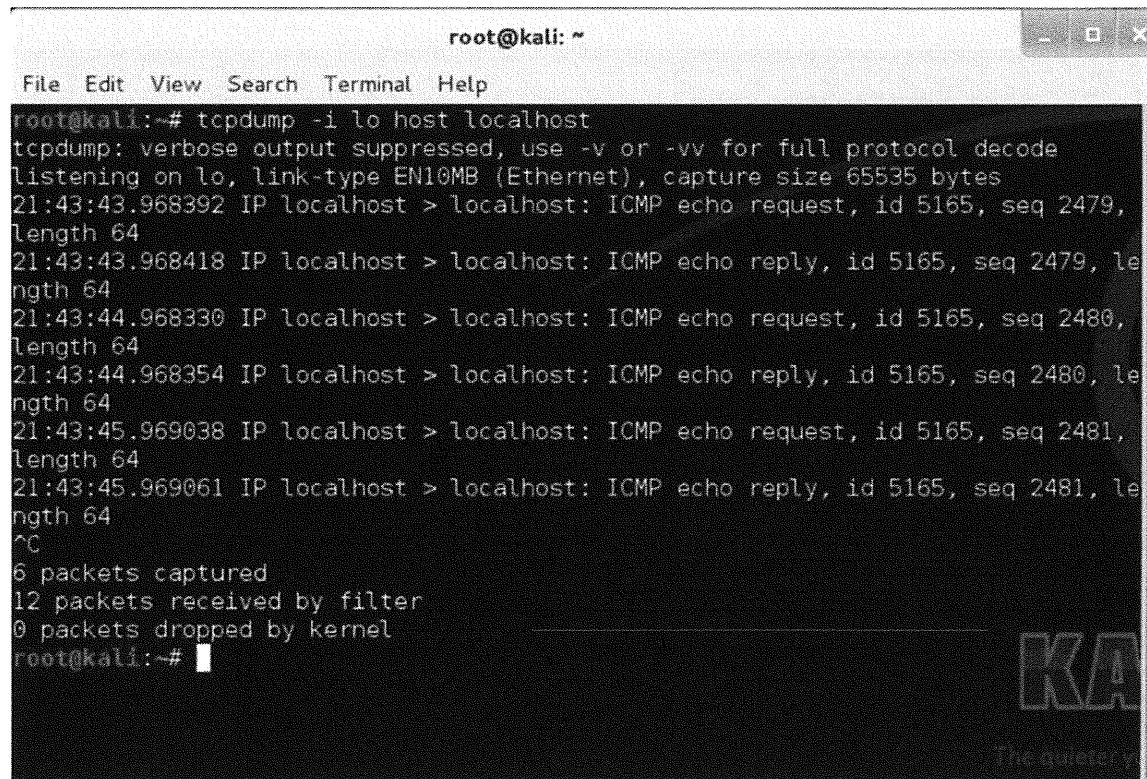
You also have options that are in plain English and make constructing meaningful arguments easier. These are shown in the following sections.

host Option

The host option allows you to specify a host in order to capture only packets that are destined to or arriving from it.

To try this option, type:

tcpdump -i lo host localhost



```
root@kali:~# tcpdump -i lo host localhost
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
21:43:43.968392 IP localhost > localhost: ICMP echo request, id 5165, seq 2479,
length 64
21:43:43.968418 IP localhost > localhost: ICMP echo reply, id 5165, seq 2479, le
ngth 64
21:43:44.968330 IP localhost > localhost: ICMP echo request, id 5165, seq 2480,
length 64
21:43:44.968354 IP localhost > localhost: ICMP echo reply, id 5165, seq 2480, le
ngth 64
21:43:45.969038 IP localhost > localhost: ICMP echo request, id 5165, seq 2481,
length 64
21:43:45.969061 IP localhost > localhost: ICMP echo reply, id 5165, seq 2481, le
ngth 64
^C
6 packets captured
12 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

and ()

These commands allow you to print captured packets that are between the first listed host and any of the hosts within the brackets. This option allows you to see specific traffic between multiple devices, instead of between only two devices.

To try this option, type:

tcpdump -i lo 'host localhost and (localhost or 192.168.1.2)'

The screenshot shows a terminal window with the title bar "root@kali: ~". The window contains the following text:

```
root@kali:~# tcpdump -i lo 'host localhost and (localhost or 192.168.1.2)'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
21:44:32.968000 IP localhost > localhost: ICMP echo request, id 5165, seq 2528,
length 64
21:44:32.968024 IP localhost > localhost: ICMP echo reply, id 5165, seq 2528, length 64
21:44:33.967992 IP localhost > localhost: ICMP echo request, id 5165, seq 2529, length 64
21:44:33.968015 IP localhost > localhost: ICMP echo reply, id 5165, seq 2529, length 64
21:44:34.968041 IP localhost > localhost: ICMP echo request, id 5165, seq 2530, length 64
21:44:34.968064 IP localhost > localhost: ICMP echo reply, id 5165, seq 2530, length 64
^C
6 packets captured
12 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

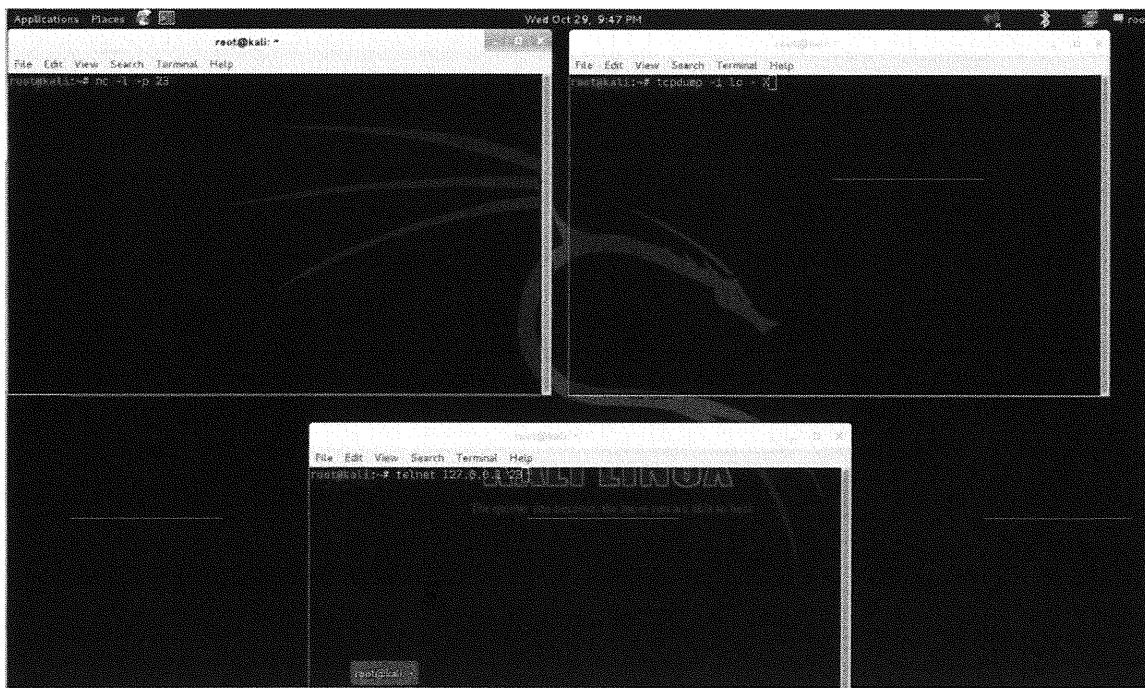
Now spend a few minutes combining these different command options. You can customize tcpdump to look and do anything you need with regard to sniffing TCP/IP-based packets.

As you can see, you can create many different configurations using the different flags and options available through tcpdump.

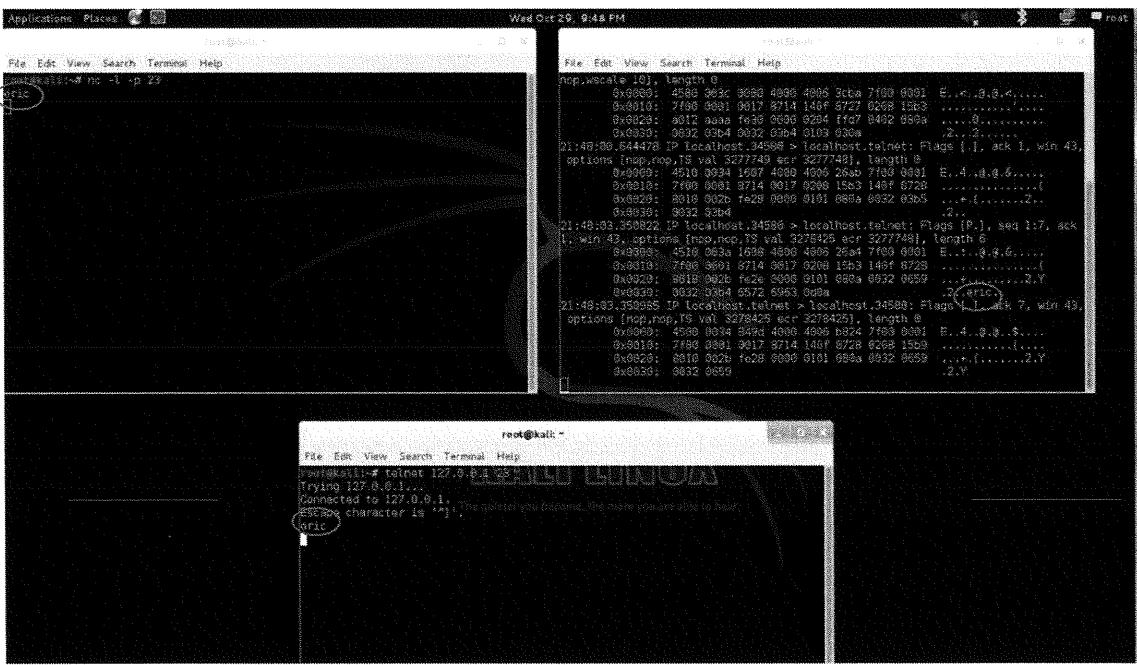
Advanced

For advanced users who want to see how a sniffer can be used to capture traffic going across a network, we are going to transfer some traffic with netcat and sniff the traffic with tcpdump. Netcat is a tool that allows you to listen on ports and transfer traffic across a network. For this exercise, you need to stop the ping command from running on the left window and stop tcpdump from running in the right window by pressing *CTRL-C* in both windows. You also need to open a third terminal window for this part of the exercise. You should now have three terminal windows open on your virtual machine.

In the window on the left, from the command prompt, type **nc -l -p 23** and press *Enter* to simulate a telnet service. In the right window, type **tcpdump -i lo -X** to capture the ASCII of the traffic. In the third window, type **telnet 127.0.0.1 23** and press *Enter* to connect you to the netcat listener.



In the bottom window in which you are running netcat, type **eric** and press *Enter*. The word “eric” is transferred across the screen. Look at your tcpdump window closely to see the words “eric.” With any protocol that sends the information in plaintext, such as telnet and FTP, all of the traffic can be sniffed.



This example shows how a sniffer can be used to capture any information going across the network. For network defenders, this can be a powerful tool for troubleshooting a network and looking for anomalous traffic. Anything that can be used for good can be used for evil. An attacker can also use a sniffer to capture passwords and other sensitive information.

Exercise: tcpdump

1. Can tcpdump collect UDP packets?
2. Can tcpdump be run as a GUI?
3. What flag or option would you use to specify a particular interface?
4. Can you mix multiple flags or options in a single command line?
5. What is the Windows equivalent for tcpdump?
6. Is it faster to have tcpdump resolve host names or use the corresponding IP addresses?

SANS Security Essentials - © 2016 Secure Anchor Consulting LLC

Exercise: tcpdump

The following questions are answered in the following section:

1. Can tcpdump collect UDP packets?
2. Can tcpdump run as a GUI?
3. What flag or option do you use to specify a particular interface?
4. Can you mix multiple flags or options in a single command-line?
5. What is the Windows equivalent for tcpdump?
6. Is it faster to have tcpdump resolve host names, or use the corresponding IP addresses?

Exercise Solutions: tcpdump

1. Yes
2. No
3. -i <interface>
4. Yes
5. windump
6. Using default or raw IP addresses is faster.

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

Exercise Solutions: tcpdump

The following are the answers to the previous set of questions:

1. Yes, tcpdump can collect UDP packets.
2. No, tcpdump does not run as a GUI.
3. To specify an interface, use the -i<interface>.
4. Yes, you can mix multiple flags or options in a single command-line.
5. Windump is tcpdump's equivalent.
6. It is faster to use default or raw IP addresses to resolve host names.

tcpdump Summary

- tcpdump is a good sniffer for learning about networks and basic troubleshooting
- It does not show entire packets (by default), and you have to do manual decodes for payloads
- There are more powerful sniffers, but the price is right for learning

SANS Security Essentials - © 2010 Secure Anchor Consulting, LLC

tcpdump Summary

This section intentionally left blank.

SECURITY 401 - SANS

Security Essentials

The End

SANS Security Essentials - © 2016 Secure Anchor Consulting, LLC

This page intentionally left blank.