I'm not robot
reCAPTCHA
Continue

Building virtual machine labs: a hands-on guide pdf full

|||||||||||||||||| |||||||||||||||||| |||||||||||||||||| |||||||||||||||||| Building Virtual Machine Labs A Hands-On Guide by Tony Robinson Building Virtual Machine Labs. Copyright © by Tony Robinson All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means electronic or mechanical, including photocopying, recording, or by any information or retrieval system, without the prior written permission of the copyright owner and the publisher. ISBN-13: 978-1546932635 ISBN-10: 1546932631 Cover Illustration: Tony Robinson Editor, Technical Reviewer: Lord Gunter Product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, we are using the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark. The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of this work, the author shall not have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in it. I would like to thank several people for helping me to see this book through to the end. First and foremost, my employer, Hurricane Labs – If they didn’t allot me the time to write this, I would have never gotten this far. Next, I would like to thank “Lord Gunter” for his herculean effort in both proofreading this massive work, in addition to his efforts spent editing. He worked on this guide with me for over three months of his own volition because he believed in this book and |||||||||||||||||| |||||||||||||||||| what it could mean for IT novices and veterans alike, around the world. Next, I want to thank my wife. She got to listen to me swear, curse and rant whenever anything went wrong while I was writing this monstrous book. I also want to say thanks to my friends over at the RallySecurity podcast for allowing me to both promote and vent about the book writing process numerous times while podcasting/streaming. Last but not least, you. Without you, I wouldn’t have had the will to go on. It is because of you that I persisted in my efforts and decided to see this all the way through. Table of Contents 1 Purpose of this Book ...13 1.1 A Note About Software Versions ...15 2 Prerequisite Knowledge ...16 3 Hypervisor and Hardware Considerations ...18 3.1 Introduction to Virtualization...18 3.2 Introduction to Hypervisors ...20 3.3 What is a Hypervisor? ...20 3.4 Bare-metal Hypervisors ...20 3.5 Hosted Hypervisors ...22 4 Hardware Considerations ...23 4.1 RAM as a Performance Factor ...23 4.2 Disk I/O as a Performance Factor ...23 What is seek time? |||||||||||||||||| |||||||||||||||||| ...24 4.3 CPU Cores and Features as a performance Factor...24 4.4 Performance is a Vicious Cycle...25 5 Understanding Virtual Networks - Hosted vs. Bare-metal Hypervisor Networking ...26 5.1 Hosted Hypervisor Networking - Host-Only, Bridged, and NAT Network Segments ...26 5.2 Bridged Networking ...26 5.3 NAT Networking ...26 5.4Host-Only Networking...27 5.5 Virtual Network Adapters and You...27 5.6 Bare-metal Hypervisor Networking - Virtual Switches...27 6 Lab Overview ...29 6.1 Design ...30 6.2 Lab Network Description...31 6.3 Bridged Network ...31 6.4 Management Network ...31 6.5 IPS 1 and IPS 2 Networks ...31 6.6 AFPACKET Bridging between IPS 1 and IPS 2 ...32 6.7 Why All The Trouble? ...32 7 VMs, Resource Allocations, and Minimum Hardware Requirements ...34 |||||||||||||||||| |||||||||||||||||| 8 Hypervisor Guides...35 9 Setup - Microsoft Client Hyper-V ...37 9.1 Installation ...37 9.2 Hypervisor Preferences ...41 9.3 Server Settings ...42 9.4 User Settings ...46 9.5 Virtual Switches ...47 9.6 Virtual Switch Types ...47 9.7 Creating Virtual Switches Using the Virtual Switch Manager ...48 9.8 Creating the First VM, pfSense ...52 9.9 Adding a New VM ...52 9.10 Initial VM Settings ...60 9.11 Installing pfSense...65 9.12 Final VM Settings ...70 9.13 Network Configuration ...73 9.14 webConfigurator - Initial Setup ...77 9.15 Making Checkpoints ...79 9.16 pfSense Summary ...81 9.17 What’s |||||||||||||||||| |||||||||||||||||| Next?...82 9.18 Final Connectivity Checks and Troubleshooting ...82 9.20 Your Turn ...85 9.21 Kali Linux VM ...86 9.22 SIEM VM...88 9.23 IPS VM...90 9.24 Metasploitable 2...93 9.25 Port Mirroring and MAC spoofing ...98 9.26 Configuring the IPS VM as a Port Mirroring Destination... 100 9.27 Configuring the pfSense VM as a Port Mirroring Source ... 101 9.28 Port Mirroring for the Remaining VMs... 102 9.29 Next Steps... 102 10 Setup - Oracle VirtualBox ... 103 10.1 Installation ... 103 10.2 Hypervisor Preferences ... 103 10.3 Creating the first VM, pfSense ... 106 10.4 Adding a New VM ... 107 10.5 Initial VM Settings ... 113 10.6 Installing |||||||||||||||||| |||||||||||||||||| pfSense... 122 10.7 Final VM Settings ... 123 10.8 Network Configuration ... 126 10.9 webConfigurator - Initial Setup ... 131 10.10 Take a Snapshot ... 134 10.11 pfSense Summary... 139 10.12 What’s Next? ... 140 10.13 Final Connectivity Checks and Troubleshooting ... 140 10.14 Your turn... 143 10.15 Kali Linux VM ... 144 10.16 SIEM VM... 146 10.17 IPS VM... 148 10.18 Promiscuous Mode ... 151 10.19 Metasploitable 2 ... 152 10.20 Next Steps... 159 11 Setup - VMware Fusion Pro ... 160 11.1 Installation ... 160 |||||||||||||||||| |||||||||||||||||| 11.2 Hypervisor Preferences ... 160 11.3 Creating the First VM, pfSense... 165 11.4 Adding a New VM ... 166 11.5 Installing pfSense... 183 11.6 Final VM Settings ... 186 11.7 Network Configuration ... 187 11.8 webConfigurator - Initial Setup ... 190 11.9 Take a Snapshot... 194 pfSense Summary ... 197 11.10 What’s Next? ... 198 11.11 Final Connectivity Checks and Troubleshooting ... 198 11.12 Your Turn ... 201 11.13 Kali Linux VM ... 202 11.14 SIEM VM... 204 11.15 IPS VM... 206 11.16 Metasploitable 2 ... 209 11.17 Next Steps... 210 |||||||||||||||||| |||||||||||||||||| 12 Setup - VMware Workstation Pro ... 211 12.1 Installation ... 211 12.2 Hypervisor Preferences ... 212 12.3 Virtual Networks... 214 Creating the First VM, pfSense ... 218 12.4 Adding a New VM ... 219 12.5 Installing pfSense... 232 12.6 Final VM Settings ... 236 12.7 Network Configuration ... 238 12.8 webConfigurator - Initial Setup ... 241 12.9 Take a Snapshot... 244 12.10 pfSense Summary... 246 12.11 What’s Next? ... 247 12.12 Final Connectivity Checks and Troubleshooting ... 247 12.13 Your Turn ... 250 12.14 Kali Linux VM ... 251 12.15 SIEM VM... 253 12.16 IPS |||||||||||||||||| |||||||||||||||||| VM... 256 12.17 Metasploitable 2 ... 259 12.18 Next Steps... 261 13 Setup - VMware vSphere Hypervisor (ESXi) ... 262 13.1 Installation ... 262 13.2 Accessing ESXi... 265 13.3 Hypervisor Setup ...268 13.4 Licensing... 268 13.5 Networking and Virtual Switches ... 268 13.6 Creating Virtual Switches ... 269 13.7 Port Groups... 273 13.8 Adding Port Groups via the ESX Web Interface... 274 13.9 Resolving Some Web Interface Bugs ... 275 13.10 VMware Flings... 275 13.11 What if I don’t want to use experimental software? ... 281 13.12 Final Flight Check ... 287 13.13 Creating the First VM, pfSense... 289 |||||||||||||||||| |||||||||||||||||| 13.14 Adding a New VM ... 293 13.15 Installing pfSense... 301 13.16 Final VM Settings ... 305 13.17 Network Configuration ... 307 13.18 webConfigurator - Initial Setup ... 311 13.19 Take a Snapshot ... 314 13.20 pfSense Summary... 316 13.21 What’s Next? ... 317 13.22 Final Connectivity Checks and Troubleshooting ... 317 13.23 Your Turn ... 320 13.24 Kali Linux VM ... 321 13.25 SIEM VM... 323 13.26 IPS VM... 325 13.27 Metasploitable 2 ... 327 13.28 Next Steps... 336 14 pfSense Firewall Rules and Network Services Guide... 337 14.1 Firewall Rule Configuration - Hosted Hypervisors ... 337 |||||||||||||||||| |||||||||||||||||| 14.2 Firewall Rules for the Bridged Network ... 337 14.3 Firewall Rules for the Management Network ... 338 14.4 Firewall Rules for the IPS Network... 340 14.5 Firewall Rule Configuration - Bare-metal Hypervisors ... 342 14.6 Firewall rules for the Bridged Network... 342 14.7 Firewall Rules for the Management Network ... 343 14.8 Firewall Rules for the IPS Network... 344 14.9 Network Configuration - Core Network Services ... 346 14.10 NTP... 346 14.11 DHCP ... 348 14.12 DNS Resolver ... 350 14.13 Squid Proxy ... 352 15 Defense in Depth for Windows Hosted Hypervisors ... 355 15.1 Unbinding Network Protocols on Windows Virtual Adapters ... 356 15.2 Using Windows Firewall to Limit Exposure of Windows Hypervisor Hosts ... 361 16 Automated Patching for Linux Lab VMs ... 370 16.1 updater.sh ... 370 17 Remote Lab Management... 372 |||||||||||||||||| |||||||||||||||||| 17.1 Windows Remote Access ... 372 17.2 Persistent Static Routes ... 372 17.3 Windows SSH and SCP Software ... 375 17.4 Generating an SSH key in Windows using PuTTYgen ... 376 17.5 Using mRemoteNG - Connection Files... 386 17.6 Using mRemoteNG - PuTTY Saved Sessions ... 391 17.7 Enabling Key-Based Authentication on Linux/Unix systems ... 396 17.8 Key Copy Method 1: echo append to authorized_keys ... 397 17.9 Key Copy Method 2: usingvi ... 398 17.10 Key Copy Method 3: SCP... 400 17.11 Making sure it worked... 404 17.12 How to use Key-Based Authentication with WinSCP... 405 17.13 Linux, BSD, and OS X Remote Access... 409 17.14 Static Routes in Linux and OS X ... 409 17.15 Adding Routes to Linux with the ip Command ... 409 17.16 Adding Routes to OS X/BSD with theroute command ... 410 17.17 Making Static Routes Persistent... 411 17.18 Linux and BSD Route Persistence via /etc/rc.local ... 411 17.19 OS X Route Persistence with Hosted Hypervisors |||||||||||||||||| |||||||||||||||||| ... 413 17.20 flightcheck.sh ... 414 17.21 OS X route persistence for Bare-metal Hypervisors ... 417 17.22 flightcheckBM.sh... 417 17.23 The ssh and scp terminal Applications ... 419 17.24 iTerm2 and Terminator... 420 17.25 Generating ssh keys using ssh-keygen ... 422 17.26 The alias Command ... 424 17.27 Enabling Key-Based Authentication in Unix/Linux Systems ... 428 17.28 Key Copy Method 1: echo append to authorized_keys ... 429 17.29 Key Copy Method 2: usingvi ... 432 17.30 Key Copy Method 3: SCP... 435 17.31 Making Sure it worked ... 437 17.32 Using key-based authentication with the SCP command ... 440 17.33 How to Enable SSH on Kali Linux ... 442 17.34 Enabling, and securing root SSH ... 448 17.35 Adding your SSH public key to root’s authorized_keys file... 449 17.36 Disabling password authentication entirely via sshd_config ... 453 18 Network Design Factors When Working with bare-metal Hypervisors ... 456 |||||||||||||||||| |||||||||||||||||| 18.1 Prereqs... 458 18.2 Creating Static Routes... 459 18.3 Creating Firewall Rules ... 460 18.4 Dealing with DHCP... 464 18.5 Jump Boxing... 465 18.6 Using a Raspberry Pi as a Jump Box ... 467 18.7 Installing the Raspbian Image to your Raspberry Pi ... 467 18.8 Configuring Raspbian ... 473 18.9 Creating a Jump Box VM ... 478 18.10 Other Physical Jump Boxes ... 484 18.11 Preparing Your Jump Box for Service ... 485 18.12 Configuring Static DHCP Address Allocations ... 485 18.13 Enabling Key-Based Authentication for your Jump Box ... 486 18.14 Windows ... 486 18.15 Linux/OS X/BSD ... 489 18.16 Adding Static Routes to your Jump Box... 491 18.17 Adding Firewall Rules and SSH tunnels to allow access to the VM lab networks 492 |||||||||||||||||| |||||||||||||||||| 18.18 I Can Still Access the pfSense webConfigurator with my Management Workstation ... 493 18.19 I Have Lost Access to the pfSense webConfigurator UI ... 495 18.20 TCP Forwarding and You ... 499 18.21 Windows SSH Tunnels ... 500 18.22 Linux/BSD/OS X SSH Tunnels ... 507 18.23 Testing your Dynamic Tunnels with FoxyProxy ... 510 18.24 Troubleshooting Dynamic Tunnels... 516 18.25 Testing Your Forward Tunnels ... 520 18.26 Windows ... 520 18.27 Linux/OS X/BSD ... 524 18.28 Understanding SSH Tunnels... 527 18.29 Closing Notes on Jump Boxing ... 528 18.30 Key-Based Authentication: Managing SSH Keys for Tunneled Connections ... 528 19 IPS Installation Guide... 530 19.1 Installing and configuring Snort (via Autosnort) ... 530 19.2 Installing and configuring Suricata (via Autosuricata) ... 536 19.3 Testing your IPS Bridge... 541 20 Splunk Installation Guide |||||||||||||||||| |||||||||||||||||| ... 544 20.1 Initial Setup (Server Installation) ... 545 20.2 (Optional) Requesting and Implementing a Splunk Dev License ... 556 20.3 Universal Forwarder Setup ... 560 20.4 Splunk TA for Suricata ... 563 20.5 Hurricane Labs Add-On for Unified2... 568 20.6 Starting The Forwarder + Persistence... 572 20.7 Testing Splunk and the Universal Forwarder ... 575 20.8 Generating The Test Battery ... 576 20.9 Verifying Results with Snort ... 584 20.10 Verifying Results with Suricata ... 587 21 In Your Own Image ... 590 21.1 Visions of What Might Be... 590 21.2 Malware Analysis Lab ... 591 21.3 Penetration Testing Lab ... 594 21.4 IT/OPs Lab...596 22 Summary ... 598 22.1 What Have We Learned Today?... 598 23 Epilogue: We Need You (Now More than Ever) |||||||||||||||||| |||||||||||||||||| ... 600

1 Purpose of this Book This guide is designed to teach you about virtualization and how to build out the virtual machine lab environment that is easy to maintain, portable, relatively well secured, and flexible enough to accommodate IT and security students that need an environment to practice their trade. The goal is to teach you how to build the baseline network and get familiar with using a hypervisor of your choice. The initial network and VM (Virtual Machine) design we will be working on together can easily be expanded upon, or swapped out to support various roles, such as: ● Testing and/or developing new systems administration tools ● Learning the ropes for offensive security tools for red team ● Practicing with detection and response tools for blue team ● Providing a safe, secure environment to perform reverse engineering, malware analysis and/or exploit development with reasonably good security protections in place This guide is not meant to be read from front to back. If you do this, you are going to get really bored, and notice a lot of repetition. Think of this book as a “Choose your own adventure” novel; this book covers how to produce a robust virtual machine lab environment across five different hypervisors. Unless you’re crazy, a VM enthusiast, or a researcher (or some combination), the likelihood that you will want or need to read all five hypervisor setup guides is going to be pretty low. Keeping that in mind, here are my recommendations: Read all the chapters up to, and including the Hypervisor Guides chapter in order to develop a better understanding of the skills you’ll want and need to create your own lab, better understand how virtual networks work in general, hardware recommendations, and finally, understand what you are building, before you pick a hypervisor and actually start building your lab environment. There are then five chapters detailing how to perform initial setup and configuration of five unique hypervisors: |||||||||||||||||| |||||||||||||||||| ● Oracle VirtualBox ● Microsoft Client Hyper-V ● VMware Workstation Pro ● VMware Fusion Pro ● VMware ESXi These chapters instruct you on how to acquire and install the hypervisor of your choosing, configure the hypervisor and VMs to support the virtual machine lab environment I will teach you how to build, and perform initial installation and setup of those virtual machines. Choose a hypervisor that suits your budget and your goals, and follow the setup instructions. After performing the setup and configuration tasks for the hypervisor of your choice, you are then meant to finish configuring the virtual machines in the lab environment, as well as the hypervisor host or management workstation you will be using to access your virtual machines. Each of the hypervisor setup guides has a section entitled Next Steps that will guide you on recommendations on what tasks need to be done to finish making the lab functional (e.g. the IDS and Splunk installation chapters), as well as what supplemental chapters to consider reading (e.g. enabling remote access for the lab VMs, hardening hosted hypervisors on Windows, network design factors for bare-metal hypervisors, etc.) for a much better experience when utilizing your newly built lab environment. 1.1 A Note About Software Versions Writing books for security and/or most IT disciplines is a daunting task. The moment you put ink to paper, the information contained in the book deprecates. You see this a lot with textbooks where there are multiple revisions that need to be written to discuss updates in the material. I’ll make mention of what software version for both hypervisors and operating installation ISOs I used throughout the guides, but don’t obsess over using the exact same version I used when I made these guides. These are merely the software versions I had available to me while writing this book. As a security practitioner, I always recommend updating your software when updates are available, and using the most current software version available. This includes hypervisors and OS distributions. |||||||||||||||||| |||||||||||||||||| If you’re from the future and using future versions of hypervisors, and future ver sions of operating systems, you may notice that some configuration settings may or may not be in the exact place a screen capture I made said it was going to be in. Button colors and styles may have changed, radio buttons may now be checkboxes, etc. This is because UI (User Interface) developers may or may not have changed exactly where a given configuration setting is. This is just a fact of life when it comes to new software releases. Sometimes they do it because they can’t leave well enough alone, or sometimes they just want to make the user experience (UX) for their product better. So if you’re panicking because a given configuration setting has moved, or a checkbox isn’t in the location indicated by my screen captures or instructions indicated they should be, don’t panic. This is the first rule of any IT related discipline. The second rule is that software changes. Sometimes arbitrarily, sometimes for the better. The third rule is to consult the documentation. Maybe the configuration option has migrated to a new menu location, or maybe it was integrated as part of another, related setting. Consult the product patch notes, documentation included with the software, and/or online knowledgebase/forums for the product to find out where the configuration setting lives now. The goal of this book isn’t to mindlessly instruct you to click here, open this menu, and check these boxes, it is also for you to understand WHAT the configuration settings that you are modifying do, and WHY I am telling you to modify them so that if and when you want to experiment, make changes, and add or remove features to your lab, you have the knowledge and proper understanding what the settings do as opposed to what buttons you need to press. 2 Prerequisite Knowledge Here are some things you should have a basic to intermediate grasp of in order to get the most out of this guide: ● TCP/IP Networking ○ Understand what an IP address is as well as how to configure an IP address, default gateway and/or DNS servers in Windows as well as Linux/Unix-based operating systems. (e.g. network adapter properties in Windows, ifconfig, ip link, ip route, route, /etc/resolv.conf, etc.) |||||||||||||||||| |||||||||||||||||| ○ Basic understanding of RFC1918 addressing ○ Basic understanding of subnetting ○ Understand the Open Systems Interconnection (OSI) model and/or TCP/IP networking models ○ A basic understanding on how stateful firewalls operate, and on what criteria they typically block (e.g. IPv4/IPv6 addresses, transport protocol (tcp/udp/icmp) and port/icmp code (e.g. port 80/tcp, port 123/udp, icmp type 0)) ○ Familiarity on ports and protocols (e.g. is it TCP or UDP) for common network services (e.g. FTP, HTTP/HTTPS, NTP, SSH, DNS, etc.) ● Familiarity with Operating Systems and their Installation Procedures ○ Familiarity with installing Linux/Unix/Windows operating systems is absolutely necessary. If you know what an ISO image is, and how to boot from one, this will help you progress a bit faster ○ Familiarity with the Unix/Linux/Windows command line (e.g. shells) is key; most of the lab is based on Linux and BSD systems that will NOT have a GUI installed. You’ll need to know how to navigate the shell to navigate the file system, edit configuration files, utilize CLI tools, and run shell scripts. ■ You should know how to perform various network troubleshooting commands from the command line on Linux/OS X/BSD systems, as well as Windows systems (e.g. ping, netstat, wget, curl, ipconfig, ifconfig, etc.) ■ You should be familiar with at least one command line text editor that Linux/OS X/BSD systems use (e.g.vim, nano, emacs, etc.), or if you insist on writing/editing your config files on Windows, then copying them to lab VMs, you should be familiar with the dos2unix command to prevent Linux systems from failing to parse config files properly ■ Familiarity with SSH and SCP for remotely managing and copying files to Linux/OS X/BSD systems will be extremely helpful ● Virtualization ○ If you are familiar with the basics of virtualization, it will make a lot of this |||||||||||||||||| |||||||||||||||||| guide much easier to understand. There are numerous resources out there for learning the basics I’ve listed. If you understand most of the points listed above, you shouldn’t have too hard of a time following along. If you don’t... things might be difficult at first, but the only way they will get easier is if you struggle along the way. If you are not against reading a good book or two, the publishing company No Starch Press produces a variety of high quality reference books to help you gain a better understanding of various subjects; of particular note are the books The Linux Command Line, TCP/IP Guide, Network Know-How, and Practical Packet Analysis. If you’re looking for free resources, the website cybrary.it has a great collection of free videos. Take a look at some of the videos produced for COMPTIA certifications, specifically the Linux+, A+ and Network+ coursework, as they have direct application to what we’re doing here, and will teach enough of the basics to allow you to progress a bit easier. Additionally, Zed Shaw has created a series of guides collectively called Learn Programming The Hard Way. He has a nice crash course introduction to the command line for various operating systems here: . 3 Hypervisor and Hardware Considerations Before you get started with the hypervisor setup guides, it helps to know some of the basics of virtualization. This is going to involve you learning about hardware resources hypervisors need, how VMs actually work, virtual networking, and finally, understand how your hypervisor choices will influence your lab’s network design. Before we dive into the hypervisor guides, I’m going to give you the crash course on what virtualization is, different types of hypervisors, how virtual networking works on different hypervisors, illustrate how the lab network we will be created, then instructing you on how to recreate the lab network and VMs on a variety of hypervisor platforms. |||||||||||||||||| |||||||||||||||||| 3.1 Introduction to Virtualization Virtualization is the concept of taking one physical computer and splitting its resources into separate chunks/containers in order to host smaller, independent virtual systems using those reserved resources. For example, if you have a physical machine with 2 CPU cores, 4GB of RAM, and 40GB of disk space, you could, using special software called a hypervisor, allocate a portion of these resources to create a container to host a virtualized computer. You could install any OS you have licensing for and have it using that chunk of disk, CPU and memory you set aside. These virtual machines, for all intents and purposes, are independent computers. You can host many virtual machines on a single physical computer, with the only limitations being the amount of resources the host computer has, and how many resources the virtualized computers (known as virtual machines, or VMs) require to run. Note: In the screen capture above for example, there are three VMs listed. The VM that is highlighted is named “pfsense”. In this case, I set aside 512MB of |||||||||||||||||| |||||||||||||||||| ram, 5GB of hard drive space, and 1 virtual CPU to install and run the pfSense OS (operating system) in a VM. Virtualization became a big deal because it allows companies to run more services, and do more things with less physical hardware; not to mention it's easier for developers, researchers, and IT professionals to have a testing environment full of virtualized systems for testing compatibility, patch deployments, and a ton of other tasks that would have required a lot of physical hardware. Not only that, virtualization software supports snapshots or the ability to capture the status of a virtualized system at a certain point in time, and restore the virtual machine to that state at will. This means that test cases, malware analysis and several other scenarios that make a significant number of changes to the system's operating status can be undone in mere moments, restoring the VM to a known good configuration state with the click of a button. 3.2 Introduction to Hypervisors Now that you know what a VM is, let's talk about hypervisors. There are a wide variety of them to choose from. Your first hurdle is choosing which option is right for you, and that depends on a number of different factors. 3.3 What is a Hypervisor? A hypervisor is software that is used to create, manage, and run, virtual machines. Hypervisors are responsible for resource allocation (CPU, disk, RAM), network configuration, virtual hardware allocation, snapshot management, and controlling the current operating status of VMs that they are responsible for managing. There are two classes of hypervisors recognized today: bare-metal hypervisors, and hosted hypervisors. 3.4 Bare-metal Hypervisors A bare-metal hypervisor is installed directly onto server hardware (in IT/sysadmin circles, known as big iron or the bare metal - hence the name for this class of hypervisor). Bare-metal Hypervisors usually feature a very lightweight operating system with minimal functionality in addition to their hypervisor functionality. In fact, in most cases, bare-metal hypervisor |||||||||||||||||| |||||||||||||||||| installation can be done on an SD card or USB drive, leaving the internal hard drives and/or Solid State Drives (SSDs) 100% dedicated to the virtual machines. Once the bare-metal hypervisor is installed, booted, and given a network configuration, typically it is managed remotely over the network through a web interface, API, or through some sort of an application installed on a workstation you use to manage the hypervisor remotely. For sake of simplicity, throughout this guide, I will refer to this system as a management workstation. All the major configuration aspects for a bare-metal hypervisors are done from the management console/web interface, using a management workstation. While bare-metal hypervisors tend to be very lean, they are packed with advanced features with more of a focus towards enterprise environments that require those features for a production environment where downtime means lost revenue. These are features such as fault tolerance, availability/failover solutions, advanced networking, etc. Since bare-metal hypervisors are commonly used in enterprise networks all over the world, it would definitely benefit you to learn how to use them in lab environment, and making yourself familiar with how they operate. However, be aware that some bare-metal hypervisors are EXTREMELY picky about what hardware they work with (VMware vSphere Hypervisor for example, is notorious for being extremely picky with what hardware it will detect and use). Even then, even if you can actually get the hypervisor to recognize your system’s hardware, not all of your hardware’s features may be supported. For example, integrated network ports on a system motherboard may not be supported, or a built-in RAID controller may not be recognized by the hypervisor. This is something you either have to deal with, unfortunately. Either you will need to find supported hardware, or find another baremetal hypervisor that isn’t so picky about what hardware it supports. Popular bare-metal hypervisors include Hyper-V (Client and Server edition), Citrix Xenserver, Proxmox, and VMware vSphere Hypervisor (also known as ESXi). |||||||||||||||||| |||||||||||||||||| Note: The illustration is VMware vSphere Client, which is used to manage ESXi bare-metal hypervisors remotely. The newest versions of ESXi also feature a built-in web interface for system management as well. 3.5 Hosted Hypervisors Hosted hypervisors differ from bare-metal hypervisors in that they are an |||||||||||||||||| |||||||||||||||||| application that is installed on top of an already installed operating system, such as a Linux, OS X, BSD or Windows. Hosted hypervisors are applications that are hosted by the installed operating system, thus the name. Typically, some sort of a console is installed as a part of the hosted hypervisor to allow users logged on to the workstation to manage the configuration locally, though some hosted hypervisors have remote management features that can also be used. Popular hosted hypervisors include VMware Workstation, VMware Fusion, and Oracle VirtualBox, among others. Note: The illustration depicted above is Oracle VirtualBox. VirtualBox is a free, multi-platform hosted hypervisor. 4 Hardware Considerations What hardware do you have available to run a VM lab? Do you plan on using a spare desktop you had lying around? Your primary workstation? A laptop? An |||||||||||||||||| |||||||||||||||||| actual server? The hardware you have available to dedicate to your lab environment will determine how far you can take your lab, and what type of hypervisor you will be using (e.g. bare-metal vs. hosted). In this section, we will be discussing how RAM, Disk I/O, number of CPU cores, CPU featureset, and motherboard BIOS dictate how well your hypervisor and VMs will perform. 4.1 RAM as a Performance Factor All operating systems, applications, utilities and VMs all require some amount of system memory to perform their tasks. Obviously, the more you have, the more applications and VMs with their own applications you can run concurrently. While there are some design factors in hypervisors that allow RAM that is not being used in one VM to be allocated to another VM when it is needed, you should never rely on these technologies, because it could result in overextending your RAM, which leads to swapping/paging to disk, which results in disk thrashing, which leads to disk performance problems, further exacerbating your performance problems. The only solutions to a lack of RAM are to reduce the number of running applications and services to decrease the demand, or to increase the amount of RAM installed on your system and/or the amount of RAM allocated to the affected VMs. 4.2 Disk I/O as a Performance Factor If at all possible, regardless of bare-metal or hosted hypervisor, you should consider using SSDs where possible. SSDs are a relatively new storage medium. They have virtually no seek time and read/write performance that is orders of magnitude faster than your traditional magnetic platter based hard drives. These performance factors are a big deal if you are doing multiple disk intensive things across your VMs at the same time. This means that the VMs will be fighting one another for I/O (input/output) access to read and write data from the same disk, which directly impacts the performance of your VMs, and if you’re running a hosted hypervisor, also affects the performance of the OS you are hosting the hypervisor on. This means that the more VMs you have on the same disk doing disk intensive tasks, the longer they all have to wait and content for disk access to read and write their data to the hard drive. |||||||||||||||||| |||||||||||||||||| If you don’t have an SSD or didn’t budget for one, don’t fret too much. Here are a few recommendations that might help you squeeze performance out of your disks and allow your VMs to coexist peacefully. ● If possible, consider installing more than one disk to running your VMs. Bare-metal hypervisors benefit by being able to spread disk I/O across multiple disks. Hosted hypervisors benefit by being able to dedicate one of the disks for OS operations, and the other for VM storage and operations. ● If possible, consider putting your disks into a RAID1 (minimum of 2 disks) or RAID5 (minimum of 3 disks) array to protect against drive failure and possibly improve disk I/O. ● If you are using a bare-metal hypervisor, consider installing it on a separate disk, USB drive, or SD card (if possible), in order to fully dedicate your disk drives solely to VM storage. ● If you are using a hosted hypervisor, minimize the amount of resourceintensive intensive tasks and applications you have running on the host OS (aside from the hypervisor of course). ● Do everything possible to avoid paging/swapping to disk on your VMs. This occurs when VMs need more RAM than the system has physically available; the VM will start using the disk like RAM. This should be avoided at all costs, even if you are using SSDs since paging/swapping causes intense wear and tear on all drive types platter or SSD (this is also known as disk thrashing). The key to preventing this is to ensure your have plenty of RAM installed, and that you have allocated enough RAM to your VMs. This means you have to monitor system performance on a regular basis. What is seek time? Seek time is the time delay associated with traditional spinning platter-based hard drives that is required for them to spin up and move tiny arms over portions of the disk platter (called read/write heads) to retrieve the data a system requires. 4.3 CPU Cores and Features as a performance Factor All services and applications have processing and calculations that need to be done to achieve some sort of input or output, or manage some aspect of system |||||||||||||||||| |||||||||||||||||| operation. All of these applications and services need some slice of time on the CPU to have their calculations performed so that they perform their functions. Hypothetically, the more CPUs/cores you have available, the more tasks a system can perform at once. Whether that is multiple tasks for a single application, or multiple tasks for multiple processes the idea is supposed to be that more cores are better. Each VM has at least 1 virtual CPU (tied to the physical CPUs on your system) attached to it, with the ability to allocate more within the limits of how many physical CPUs/cores you have installed on your PC, laptop, or server. Each VM has services and applications that need CPU time to run calculations to manage system services and input/output for running applications, just like any other physical system. The more intense the applications and services are in terms of CPU processing required, the more time and utilization those applications take to finish their calculations. This means that other applications and services need to wait for time on the CPU to perform their calculations. This in turn results in applications and services (even the entire OS in some cases) becoming unresponsive during extremely CPU intensive tasks. Like with RAM utilization, the only solutions here are to reduce the number of running applications or services, deal with the factors that are causing them to take so much CPU time, or increase the number of CPUs/cores available. In addition to the number of cores your system has available to handle CPU load, it is also important to confirm that both your CPU and/or system motherboard include support for virtualization features. Intel and AMD processors have virtualization extensions (Intel:VT-x AMD:AMD-V) that are built-in CPU features. Most of the time you can simply use your favorite search engine, and search for your processor name and the results will return links to the CPU manufacturer that usually include a spec sheet page that confirms whether or not virtualization features are included in that particular CPU model. Some of the low-end CPUs do NOT include virtualization extensions in order to cut costs, so be aware of this! In addition to checking the CPU specs, you also need to confirm that the motherboard BIOS supports virtualization. If you are using a prebuilt system from a large PC manufacturer (e.g. Dell, HP, etc.) then usually it's as easy as searching for your PC or laptop’s model name on the manufacturer’s support |||||||||||||||||| |||||||||||||||||| website to find and download a system manual, or spec sheet to confirm the features the system supports. However, if you’re like me and you’re into building your own PCs, try visiting the motherboard manufacturer’s website, search for the model of motherboard you wish to use, then download and review the motherboard documentation to confirm that virtualization is a supported BIOS feature. 4.4 Performance is a Vicious Cycle You should be seeing a pattern at this point. Don’t have enough RAM? That leads to resource contention for memory and swapping/paging to disk. Don’t have enough disk I/O? That leads resource contention for apps to read/write to and from disk. Don’t have enough CPU/cores? That leads to resource contention for calculations. All of these can feed on one another and result in poor system stability and performance as a whole. Keep an eye on CPU, disk and RAM performance metrics on a regular basis for your hosted and/or bare-metal hypervisors, and adjust accordingly. Windows systems have task manager, while most Unix-like operating systems (That is, Linux, OS X, BSD) have several performance measuring utilities that can be ran from the command line such astop, free, iostat and iotop. Most baremetal hypervisors have their own performance graphs that measure the performance of the hypervisor as a whole, as well as the performance of individual VMs in terms of disk, RAM, and CPU utilization. 5 Understanding Virtual Networks - Hosted vs. Baremetal Hypervisor Networking Let's discuss how networking is done on both hosted and bare-metal hypervisors, because this is a key factor in how you access your lab VMs, and how our lab will be built on bare-metal and hosted hypervisors. 5.1 Hosted Hypervisor Networking - Host-Only, Bridged, and NAT Network Segments In most hosted hypervisors, virtual networks are typically divided into special network segments that all serve a different function. There are usually three |||||||||||||||||| |||||||||||||||||| types of network segments that VMs get connected to: NAT, Bridged, or HostOnly network segments. Some hosted hypervisors allow you create additional custom network segments, but they usually fit into one of these three categories. 5.2 Bridged Networking Bridged network segments are used to share the host system’s network card to directly connect to the same network as the host operating system. In this case, VMs that connect to the bridged network segment act as though they are directly connected to the same physical network as the host system. From a network perspective, they look exactly like any other system connected to your physical network. They can have their own IP address on the physical network to which they are attached, and would respond to network requests like any other system on that physical network. 5.3 NAT Networking If you have an understanding on how Network Address Translation/Port Address Translation (NAT/PAT) works on a regular/home network for connecting multiple machines to the internet through a single publically routable IP address, then NAT network segments are exactly how you’d imagine them: The host operating system makes connections to external resources on behalf of the VMs connected to the NAT network. VMs connected to a NAT network share the host system’s IP address. All outbound traffic appears to be coming from the host system, and any services you set up on the VM for external access (through port forwarding) appear to be hosted on the host system as well. For the rest of you who have no idea what I’m talking about, VMs that get added to a NAT network have their own internal network addressing scheme (IP address range, subnet mask, default gateway, etc.). The default gateway for this private network that does that NAT functions is usually a special virtual network card or IP address attached to the host operating system. Any traffic not destined for other hosts in the NAT network gets routed to this address. The host operating system then sends out the request on behalf of the system on the NAT network, using its network connection. The responses (or lack thereof) are then relayed back to the VM in the NAT network. NAT networks can be used to share the host operating system’s network connection and make it |||||||||||||||||| |||||||||||||||||| appear as though the connection requests are coming directly from the host operating system, instead of directly exposing the virtual machines to the network. Some hosted hypervisors also allow you to setup port forwarding so that connections to the host OS on certain ports get forwarded to a specific VM’s IP address in the NAT network. In general, I tend to avoid using NAT network segments that most hypervisors provide. The only time I generally use NAT networks for network connectivity is if for some reason, bridged networking isn’t working properly. For instance, a bridged VM isn’t getting an IP address, or network connectivity is otherwise broken (maybe there is some sort of MAC address filtering, preventing network access, etc.). I also tend to avoid using NAT network segments because with most hosted hypervisors, the port forwarding functionality that NAT networks are supposed to provide is either buried in the user interface, or requires modifying configuration files buried somewhere in the hosted hypervisor’s installation files. 5.4Host-Only Networking Host-only networks are network segments that, by definition, do not get any sort of network connectivity to the outside world. VMs on these network segments can only communicate with hosts on the same host-only virtual network, and/or the hypervisor itself. 5.5 Virtual Network Adapters and You The NAT and Host-Only networ ks allow the hypervisor’s host OS to communicate with VMs inside of their respective networks through the use of virtual network adapters. The hypervisor creates a virtual network card and attaches it to the hypervisor host. To the host OS running the hypervisor, this virtual network card is just like any physical network card, and can be configured like one. Virtual network adapters allow your hypervisor host to connect to VMs in these networks, and interact with their network services. Some hypervisors actually allow you to disable creation of the virtual network adapter for these network segments. In the case of a NAT network, this would create a network that uses the host system for external network connectivity, but the host would not be able to directly connect to VMs behind the NAT IP |||||||||||||||||| |||||||||||||||||| address; you’d have to configure port forwarding to access any of the network services for the VMs in that NAT network. In the case of host-only networks, this creates a private internal network in which the VMs could only connect to one another, but the host OS would not be able to connect (except through the hypervisor’s console so that you could still install and configure VM on this network). Usually this is done to provide some degree of isolation between the hypervisor host and the VMs, and provide better network segmentation. 5.6 Bare-metal Hypervisor Networking - Virtual Switches Most bare-metal hypervisors implement virtual networking via virtualized network switches (vswitches). You don’t really have the concept of host-only, NAT and/or bridged networks the same way you do in hosted hypervisors. I’m going to use VMware’s vSphere Hypervisor (also known as ESXi) as an example here, since most bare-metal hypervisors (but not all) implement their virtual networking in roughly the same manner. With ESXi, you simply have virtual switches that have a number of ports you can attach virtual machines to, so that they are on the same Layer 2 network. Those vswitches can also be attached to the server’s physical network cards, allowing virtual machines attached to that switch to interact with the rest of the network through that network card. In this way, it emulates how most professional grade network switches have designated uplink ports to connect the switch to the rest of a larger enterprise network, not unlike how a hosted hypervisor’s bridged nework operates. Alternatively, you could choose not to uplink to the server’s physical network connection and the hosts attached to that vswitch would be considered isolated and only able to talk to machines attached to that particular vswitch, not unlike how a hosted hypervisor’s hostonly network operates. 6 Lab Overview In this chapter we will review a network diagram of the VM lab we will be building together, discuss resource allocations for our VMs, and discuss how our virtual network lab will be laid out, regardless of what hypervisor you use. |||||||||||||||||| |||||||||||||||||| 6.1 Design |||||||||||||||||| |||||||||||||||||| 6.2 Lab Network Description We have 5 virtual machines in our network, and 4 distinct network segments. We’re using a pfSense VM with three network interfaces to route traffic and |||||||||||||||||| |||||||||||||||||| provide security for the Bridged, Management, and both IPS networks. pfSense is a very popular, flexible and easy to use BSDbased firewall distro that, in addition to acting as a network firewall, can provide a variety of other network services. Our pfSense VM will handle routing traffic between network segments as necessary, network segmentation and security through a series of firewall rules, and core network services for our lab environment, including DHCP, DNS, and NTP services. Having pfSense sit between all of these network segments leads to why the network was designed the way it was. Each of the network segments in our lab has a particular function and reason for being there, so let’s talk about how they’re set up and why. 6.3 Bridged Network The Bridged network is like an upstream internet connection that connects our VMs to the local physical network the hypervisor is connected to, and (eventually) the internet. Firewall rules we put on the pfSense VM will be set up to prevent local hosts from accessing the lab VMs, and will prevent the lab VMs from communicating while the local hosts on the physical network, while still allowing internet access as necessary. If you’re running a bare-metal hypervisor, you will also be setting up firewall rules to allow your management workstation, or a dedicated jump box to access the VM lab systems. 6.4 Management Network The Management network acts as a trusted, safe and secure network. This network is where our SIEM (Security Intrusion Events Manager) VM will be, and one of the three network interfaces we will be using for IPS (Intrusion Prevention System) VM (I will explain why the IPS VM has 3 interfaces in just a moment), and one of the three firewall interfaces for our pfSense VM. Our IPS VM will be running either Snort ( ) or Suricata ( ) for network inspection, while our SIEM VM will be running Splunk ( ) to collect and allow us to query generated alerts from our IPS system. 6.5 IPS 1 and IPS 2 Networks |||||||||||||||||| |||||||||||||||||| The IPS 1 network hosts a Kali Linux VM, and one of our three network interfaces connected to the IPS VM, while the IPS 2 network hosts the third of 3 network interfaces for our IPS VM, as well as the Metasploitable 2 VM. Kali Linux is a very popular and easy to use penetration testing distro that is loaded with offensive security tools and scripts, while Metasploitable 2 is a very vulnerable VM. Metasploitable has been traditionally used to introduce red teamers to penetration testing and exploitation with a tool called The Metasploit Framework. In our case, we’re using Kali and Metasploitable 2 for the express purpose of testing our IPS system and making sure that these two VMs on the two network segments can communicate with one another. 6.6 AFPACKET Bridging between IPS 1 and IPS 2 Wait, so, if IPS 2 is NOT directly connected to the pfSense VM, how does Kali reach it? Remember how I mentioned our IPS VM has three network interfaces? One is connected to the Management network, while the other two are connected to both the IPS 1 and IPS 2 networks. We will be connecting the IPS 1 and IPS 2 networks together through our IPS VM, using a technology available in both Snort and Suricata called AFPACKET bridging. 6.7 Why All The Trouble? AFPACKET bridging allows the two network interfaces of the IPS VM, connected to IPS 1 and IPS 2 networks to be fused together, allowing communication between the two network segments, so long as the IPS VM is powered on, and the Suricata or Snort IPS service is running. For instance, if you wanted your lab to serve as a malware analysis lab and wanted to ensure complete isolation to the internet. If you place your VMs in the IPS 2 network, you could then turn off the IPS VM, or disable the Snort/Suricata service, and VMs in the IPS 2 network have no external access to the internet or any of the other network segments (even the IPS 1 network) whatsoever. Conversely, if you are host pentesting lab with your VMs hosted on the IPS 2 network, and you have a bunch of lab systems you wanted to update, then isolate again, you can turn on the IPS VM, bridge the two networks together, and get your network access from the pfSense interface connected to the IPS 1 network. After you’re done downloading updates, turn off the IPS VM, and you have an isolated lab environment again. This network design is called fail-closed networking. |||||||||||||||||| |||||||||||||||||| This is how it works in a nutshell. Fail-closed networking operates on the idea if that you would rather have the network be secure and unable to communicate with external networks, as opposed to fail-open, and have no protection/isolation at all. The advantages for our lab in implementing a fail-closed network are pretty great. At a moment’s notice, our lab VMs can have access to the internet, and at a moment’s notice, if there is some sort of a catastrophic issue, the VMs in the IPS 2 network can be isolated quickly. This allows us the ability to switch between the equivalent of a host-only and bridged network in an instant. To put it bluntly, it’s the secret sauce behind how our lab can be modified to accommodate a variety of needs for IT and/or information security training needs. 7 VMs, Resource Allocations, and Minimum Hardware Requirements We will be using at least 5 virtual machines as a part of the lab network we will be building together.. Here are the recommended resource allocations for the VMs:: ● pfSense: 512MB RAM, 5GB Disk, 1 cpu/core ● Kali Linux:4GB RAM, 80GB Disk, 1 cpu/core ● SIEM (Ubuntu 16.04 Server): 4GB RAM, 80GB Disk, 1 cpu/core ● IPS (Ubuntu 16.04 Server): 2GB RAM, 80GB Disk, 1 cpu/core ● Metasploitable 2: 512MB RAM, 10GB Disk, 1 cpu/core |||||||||||||||||| |||||||||||||||||| If you add up all the disk and RAM requirements, we’re looking at 255 GB of disk space to host the VMs (note that this does NOT account for space required for VM snapshots), and 11GB of RAM. As such, I would recommend a box with a minimum of 16GB of RAM, 4 CPU cores (with Intel’s VT-x or AMD’s AMD-V virtualization technology supported by the CPU and motherboard), and 500GB of disk space, at a minimum. As with most hardware requirements, more is always better. The resource allocations I ’ve recommended for the VMs ensure that each of the VMs performs well. The hardware recommendations guarantee that you have room to create an additional VM or two, as well as store snapshots of your VMs, operating system files, hypervisor files, and OS installation ISOs. If you are really in a squeeze and need to fit more VMs on the same disk, or you have less than 16GB of RAM to work with here are some things you can do to try and spread your hardware a bit further. Keep in mind that pushing your system too hard, or spreading your system resources too thin

can lead to poor performance. ● Since this is a simple lab environment, you may consider cutting the RAM allocations on the SIEM and Kali Linux VMs from 4GB to no less than 2GB. This has the potential to reduce the max RAM consumption across your VMs from 11GB to 7GB, but graphical tools you use in the Kali Linux VM may be a little sluggish, and take a little longer to load. ● You could decrease the disk space allocation on the SIEM, IPS and Kali Linux VMs from 80GB each, to 60GB each for the IPS and SIEM VMs, and 40GB for the Kali Linux VM. This has the potential to save you 20-80GB of disk space, but means you will have to closely monitor disk usage in your VMs. 8 Hypervisor Guides If you've made it this far, you know what we're building, you have the hardware to build it, and you may even have a hypervisor in mind that you want to use. I have created a guide on how to set up 5 different hypervisors to support the lab network that we want to create. Which one you choose is up to ||||||||||||||||||| ||||||||||||||||||| you. Please note that any software requirements specified below are in addition to the hardware recommendations we discussed earlier (e.g. RAM, disk, number of cores, Virtualization Extension support, etc.). Here are your hypervisor choices: Microsoft Client Hyper-V : A hosted hypervisor by the Microsoft Corporation for users of Windows 8.1 or Windows 10. You must be running Professional, Enterprise, Ultimate or Education edition, and the operating system must be 64-bit. Oracle VirtualBox: A hosted hypervisor by the Oracle Corporation. Available on practically every operating system out there. 64-bit OS is recommended. VMware Fusion Pro: A hosted hypervisor by the VMware Corporation made specifically for Apple OS X. This guide specifically requires VMware Fusion Pro edition. VMware Workstation Pro: A hosted hypervisor by the VMware Corporation for Microsoft Windows and Linux systems. 64-bit OS recommended. VMware vSphere Hypervisor (ESXi) : A bare-metal hypervisor by the VMware Corporation. Extremely picky about hardware compatibility, but free and very robust. Support for new and/or exotic systems may not be guaranteed. I have provided step-by-step instructions on how to create your VM lab on the five hypervisors listed above. Each hypervisor setup guide has the following sections: ● Initial installation of your hypervisor ● Setup and customization of the hypervisor ● Detailed step-by-step guide on creating your first VM, pfSense ○ OS installation ○ Initial network setup ○ Initial setup via the webConfigurator (the pfSense web interface) ○ Taking your first snapshot ○ Redirecting you to the pfSense Firewall Rules and Network Services guide to fully set up the pfSense VM including guidance on: ||||||||||||||||||| ||||||||||||||||||| ■ Setting up the firewall rules for the Bridged, Management and IPS networks ■ Setting up core network services ● DHCP ● DNS ● NTP ● Squid Proxy (optional) ● Guidelines to follow for creating 3 out of the remaining 4 VMs ● Instructions on how to acquire and set up Metasploitable 2 as necessary ● Recommendations on remaining chapters to read to create a fully functional lab environment such as: ○ Defense in Depth for Windows Hosted Hypervisors ○ Remote Lab Management ○ Network Design Factors When Working with Bare-metal Hypervisors ○ IDS/IPS Installation ○ Splunk Installation So without further adieu, choose a hypervisor, and get started! 9 Setup - Microsoft Client Hyper-V Note: Please be aware that this guide was written using Client Hyper-V running on Windows 10 Professional, anniversary edition. If you are running Client Hyper-V on Windows 8.x, you may notice some slight differences, but the instructions should more or less be the same. I've been u sing Client Hyper-V as my hosted hypervisor of choice recently because it's convenient (I primarily run Windows for playing video games and other applications), the hypervisor's featureset is robust, and the price is right (read: free) if you're running the right version of Windows. This guide will instruct you on how to create the our lab environment on Client Hyper-V, one step at a time. 9.1 Installation The only downside to running Client Hyper-V, is that you have to be running a premium version of Windows in order to get the option to install it. Specifically, Client Hyper-V is an additional feature you can install for the ||||||||||||||||||| ||||||||||||||||||| following Windows operating systems: ● Windows 8 (or 8.1) Pro 64-bit ● Windows 8 (or 8.1) Enterprise 64-bit ● Windows 10 Enterprise ● Windows 10 Professional ● Windows 10 Education Note: While not explicitly stated on the Windows 10 Hyper-V System Requirements page on the MSDN (bility), you'll need a 64-bit OS to be able to address all the memory we will need to use for our lab. Therefore, you'll want to use a 64-bit version of Windows 10. Additionally, any drives you use to store your virtual machines must be formatted as NTFS. I discovered this by mistake when I tried to run my VMs on an ex-FAT formatted SSD drive, and experienced errors initializing/starting my VMs. (In addition to requiring one of the premium editions of Windows 8 or Windows 10, There are certain specific hardware requirements that Client Hyper-V needs in order to run. To make it easy however, you can open up a command prompt and run: systeminfo.exe The last thing the command will output is a section labeled Hyper-V Requirements: This will tell you whether or not your hardware can run Client Hyper-V. If all the options come back Yes, you should be good to go. Otherwise, double check that your motherboard has virtualization options enabled and that your CPU supports the required virtualization options as well. To install Client Hyper-V, navigate to the Programs and Features menu, and select the Turn Windows Features on or off option. ||||||||||||||||||| ||||||||||||||||||| In the next window, Locate the Hyper-V feature, and click the checkbox so that a checkmark appears in the box, then click OK. Windows will enable Client Hyper-V and instal the management tools/console for managing the hypervisor. After the installation is complete, you will need to reboot your system. After you have rebooted, run the Hyper-V Manager application. The system will ask you if there is a remote server you wish to connect to, or if you would like to connect locally. Select the Local computer radio button and click OK. You should be greeted by a screen that looks like this: ||||||||||||||||||| ||||||||||||||||||| 9.2 Hypervisor Preferences Now that Client HyperV installed, let's do some customization. On the right pane of the Hyper-V Manager interface labeled Actions, click on HyperV Settings… to bring up the Hyper-V Settings menu for your system. You'll notice that on the left pane, there are two subsections: Server and User. The Server settings affect how the Hyper-V server operates, while the User settings determine how the client interacts with VMs on the Hyper-V console. ||||||||||||||||||| ||||||||||||||||||| 9.3 Server Settings The first server setting, Virtual Hard Disks Determines where on the system the virtual hard drives, known as vhd (virtual hard drive) or vhdx (virtual hard drive extended) files in Hyper-V terminology, will be stored. The files are the containers that Hyper-V uses for installing virtual machine operating systems and the files for that operating system. In the hardware considerations portion of this guide, we discussed the importance of disk I/O as a performance factor, but lets recap here. If you have a second physical hard drive installed on your system, and you are using a hosted hypervisor, it is a good practice to place your virtual machine files on a separate hard drive that the host operating system is NOT installed on. This is in order to maximize the amount of disk I/O available to both your VMs, and the host operating system. For example, If you have two physical hard drives installed in Windows formatted as theC: and D: drives, and you installed Windows on theC: drive (usually the default setting), you would want to store the VM files on the D: drive to make sure that your VMs and the host OS are not competing for disk I/O, impacting the performance of both the host OS and the VMs you wish to run. I chose to locate my VHD files under D:\VMs\Disks. If you change this setting, be sure to click the Apply button in the bottom right corner of the window. The second server setting, Virtual Machines, defines where the configuration files and folders for each of your VMs should be stored. While these files aren't as big as the VHD files, generally speaking, its good practice to store the virtual machine files in the same location as the VHDs for organizational purposes. In my case, I chose to store the virtual machine files in D:\VMs\Configs. The next server setting, Physical GPUs, determines whether or not virtual machines created will have direct hardware access to any installed GPUs on the host system. There should be no reason to have this functionality enabled, since none of our VMs have the means to utilize hardware like that, so you'll want to uncheck the Use this GPU with RemoteFX checkbox, then click the Apply button. ||||||||||||||||||| ||||||||||||||||||| I left the NUMA Spanning and Storage Migrations settings alone, Since they won't really have an effect on our environment. NUMA spanning is more for massive multi-CPU systems with tons of memory. It allows the system to use memory that isn't necessarily local to the CPU. Storage migration is the ability to move a number of VMs while the VMs you want to migrate are still up and running. It's a useful feature for enterprise environments, but a feature you won't have to worry about too often in a lab environment. The only time you'll be moving VMs is if you get a hardware upgrade, and chances are, you can spare the downtime in your lab if that's the case. The final server setting, Enhanced Session Mode Policy governs whether or ||||||||||||||||||| ||||||||||||||||||| not extra features of the vmconnect application will be available or not. The vmconnect application is what you use to interact with your VMs in HyperV. If you're familiar with how Microsoft's Remote Desktop Protocol application works, vmconnect operates in a very similar manner. In the RDP application, used for Remote Desktop connections, there is a tab called Local Resources that allows you to select what things, if anything, you want to have available when you connect to a remote computer You can select local drives, printers, sound devices, etc., and have them behave as though they're connected to the remote system. Enhanced Session Mode is essentially the same thing, except for the vmconnect application. If Enhanced Session Mode is allowed, the vmconnect client will be able to share client resources on the VM the user connects to. As always, there are restrictions. Enhanced Session Mode only applies to Windows Hyper-V VMs that have Remote Desktop Protocol enabled. This makes sense, seeing as how it is practically the same thing. . Since our lab is almost entirely Linux/BSD based, this has absolutely no bearing on us whatsoever. However, since the possibility exists that Windows VMs may be added to your lab network later as you need them, so I would highly recommend unchecking the Allow enhanced session mode checkbox, then clicking the Apply button to disable it. You need to be very careful about features that make resources from your hypervisor host available to VMs in your lab, because this has a habit of breaking segmentation, security boundaries, and isolation between the VM host and the VMs themselves. For instance, if you had enhanced session mode enabled, and shared drives from your host system to the VM, there is a chance that your hypervisor host could become infected with malware, or active samples could escape your lab network, limiting your ability to effectively contain it. This is why I recommend disabling enhanced session mode entirely, as a precaution. ||||||||||||||||||| ||||||||||||||||||| If you want to find out more about NUMA spanning check out this MS technet article: . If you're interested in learning more about Storage Migration, you can read this MS technet article: v=ws.11).aspx. If you'd like to learn more about Enhanced Session Mode, go here: . ||||||||||||||||||| ||||||||||||||||||| 9.4 User Settings The Hyper-V user settings are fairly selfexplanatory, but for sake of completeness, let's go over them. The Keyboard setting determines how the system should respond to special key combinations when you are using the keyboard while interacting with a VM through the console. This setting determines whether or not the host OS or the VM should interpret the keyboard shortcut you are sending. The Mouse Release Key setting defines what key combination should be entered in order to release control of the mouse from a virtual machine. Most hypervisors (hosted or bare-metal) have a way to allow the user of the hypervisor to attach to the console of a VM they are running. This would be the equivalent of attaching your keyboard, mouse, and monitor directly to the VM. If you're familiar KVM (Keyboard, Video, Mouse) switches operate, its like that. When you interact with the VM console in most hypervisors, (e.g. click anywhere in the console), the VM takes over the keyboard and mouse, and assumes that all button clicks are intended for the VM. This setting controls what key combination releases the mouse (and keyboard in most cases) from the VM console so that you can interact with other applications on the hypervisor host. The final user setting is for configuring whether or not the client will use Enhanced Session Mode, if it is available. If you followed the previous instructions, you should have already disabled enhanced session mode under the server settings. For the sake of completion, you'll want to make sure that the Use enhanced session mode checkbox is unchecked under the user settings section, then click Apply. The final option Reset Check Boxes allows you to restore all of the settings (Both Server and User settings) we configured here back to defaults if needed. 9.5 Virtual Switches So now that we have the hypervisor configured, we need to set up the network infrastructure. Hyper-V uses virtual switches to signify different virtual network segments. These virtual switches come in three varieties: External, ||||||||||||||||||| ||||||||||||||||||| Internal, and Private. 9.6 Virtual Switch Types External virtual switches allow you to use your physical host system's network connection as an uplink port to connect any VMs connected to the External virtual switch to the same physical network your host system is connected to. This would allow remote systems on your network to interact with VMs hosted on your system. External virtual switches are similar in functionality to bridged network segments that other hosted hypervisors use. Internal virtual switches create a standalone virtual network with no connection to the local network, or the internet. Hosts connected to an Internal virtual switch can only communicate with other hosts on that virtual switch, including the hypervisor host. The host system is able to talk to VMs on the Internal virtual switch through a virtual network card the hypervisor creates that is attached to the virtual switch. This virtual network card allows it the hypervisor host to communicate with VMs connected to the internal network switch over the network. As you might have guessed, Internal virtual switches are similar to host-only network segments. Private virtual switches are very similar to Internal virtual switches, in that they create a standalone network that only the hosted virtual machines can use to talk to one another directly. The main difference between an Internal and Private virtual switch is that the host system does not get a virtual network card, and therefore cannot communicate with VMs on a Private virtual switch directly. 9.7 Creating Virtual Switches Using the Virtual Switch Manager So now that you know the difference, between the three types of virtual switches, we need to create 4 of them: 1 External, 1 Internal, and 2 Private switches. In the Hyper-V Manager main screen, on the right pane labeled Actions, click on Virtual Switch Manager… to get started. ||||||||||||||||||| ||||||||||||||||||| A new window titled Virtual Switch Manager for [system name] will appear. On the right pane, you will see the label Create Virtual switch, and a window that has the choices External, Internal, and Private. Click on External so that it is highlighted, and click the Create Virtual Switch button. ||||||||||||||||||| ||||||||||||||||||| The screen will update with the new virtual switch you just created. Under virtual switch properties, there are two input boxes titled Name: and Notes:. This External virtual switch is our Bridged network in the network diagram, so name it something simple like "Bridged Network". As for notes, be sure to input notes to remind you of what function the switch serves, something like, "VM lab internet connectivity". Underneath this, you will see the section titled Connection type. If you made the wrong type of switch and need to change its type, you can do so here. Note that under external network, there is a drop-down selection. This drop-down contains a list of all the network cards installed on the physical host. This determines which network card VMs attached to this External switch will use ||||||||||||||||||| ||||||||||||||||||| for access to the physical network (and/or internet). Underneath the drop-down list, is a checkbox labeled Allow management operating system to share this network adapter. If you want the host system to be able to access the physical network through the adapter you selected, this box must be checked. After you are done configuring the Bridged Network virtual switch, click on New virtual network switch on the left pane to return to the previous screen. Create the following: Create one Internal virtual switch with the name "Management Network" and a description "VM lab segment used to administer VMs." ||||||||||||||||||| ||||||||||||||||||| Create two Private virtual switches with the names "IPS 1" and "IPS 2". IPS 1 should have the description "Connects firewall to IPS". IPS 2 should have the description "Connects IPS to vulnerable VMs".When you are done you should have four virtual switches. The Virtual Switch Manager interface should look something like this: Click the Apply button at the bottom right of the window to have Hyper-V create the new virtual switches, then click OK. 9.8 Creating the First VM, pfSense ||||||||||||||||||| ||||||||||||||||||| pfSense is the keystone holding this entire configuration together. Personally, it's my favorite firewall distro due to ease of use, the amount of functionality it includes out of the box, combined with a plugin/add-on system for additional functionality. If you have the CPU, RAM, and disk, pfSense can easily be converted into a socalled "Next-Generation" firewall. Make your way to . Download the latest installation ISO for the amd64 architecture. While you're at it, you may want to download a compression utility. The pfSense maintainers distribute pfSense as an ISO image file compressed with gzip. This means that we'll need to decompress the ISO file at some point. On Windows, I prefer 7-Zip (as my compression utility of choice for decompressing files, since 7-Zip can handle zip, gzip, rar, and 7z files (among others) easily. 9.9 Adding a New VM On the Hyper-V Manager window, on the left pane titled Actions, click New and a drop-down menu appears. Click on Virtual Machine… in the drop-down menu to start the New Virtual Machine Wizard. The fir st screen has us specify a name for our new VM. We'll keep it simple ||||||||||||||||||| ||||||||||||||||||| and name it "pfSense". This screen also has an option to allow you to store the VM files in an alternate location if you need to. Click Next. The next screen has us specify whether or not the virtual machine is a generation 1 or generation 2 VM. Without going into too many details, all of the VMs we will be creating will be generation 1 virtual machines. Click on the Generation 1 radio button, then click Next. ||||||||||||||||||| ||||||||||||||||||| Note: If you want to find out more about Generation 1 vs. Generation 2 VMs (including more OS support information) visit this page: . The next screen has us specify how much memory we want to allocate to our VM. Enter "512" in the input box labeled Startup memory: and uncheck the Use Dynamic Memory for this virtual machine. checkbox. Dynamic memory allows the virtual machine to allocate more RAM to itself as it sees fit. If you have a bit more RAM to play with, you could experiment with this feature, but for now, we're going to leave it disabled. When you are finished, click Next. ||||||||||||||||||| ||||||||||||||||||| The next screen has us configure networking. Eventually we will have 3 network interfaces in total for this VM, but for now, select Bridged Network from the Connection: dropdown. Afterwards, click Next. ||||||||||||||||||| ||||||||||||||||||| The next screen has us allocate disk space to create a virtual hard disk for to install our VM's OS and files. Select the Create a virtual hard disk radio button. Unless you have a compelling reason, the vhdx filename and location should not need to be changed. The Size: input box should be changed to "5" to allocate 5GB of disk space to this virtual machine. After altering the size of the virtual hard disk, click Next. ||||||||||||||||||| ||||||||||||||||||| Before we move on to the next screen, if you haven't already downloaded your pfSense ISO I would suggest doing so now. Please be aware that the pfSense ISO is gzip compressed and will need to be extracted/decompressed for Hyper-V to use it. To decompress your pfSense ISO using 7-Zip, open file explorer, and navigate to where you download the pfSense ISO file. Right click on the file, choose the 7-Zip from the context menu, and the Extract Here option. This will extract the gzipped ISO for you and place in the same directory as the original gzipped file you downloaded. On the next screen in the New Virtual Machine Wizard, labeled Installation Options, Click the radio button Install an operating system from a bootable CD/DVD-ROM. This will make the Media portion of the screen active. Click ||||||||||||||||||| ||||||||||||||||||| on the radio button labeled Image file (.iso): then click browse to open the file explorer and locate the unzipped pfSense ISO. After you have select the pfSense ISO, click Next. The final screen verifies the settings we used to create our first VM. Confirm that your VM's configuration looks similar to what is presented on this screen, then click Finish. ||||||||||||||||||| ||||||||||||||||||| 9.10 Initial VM Settings So now we have our first VM created. However, before we start it to install its operating system, there are a few more adjustments we have to make. In the Hyper-V Manager, under the Virtual Machines pane, right click on pfSense, and select Settings. ||||||||||||||||||| ||||||||||||||||||| In the settings window for pfSense, The first thing we're going to do is add new hardware. Specifically, we have to add two more network adapters. Under Hardware in the left pane, click on Add Hardware to highlight, then on the right pane, select Network Adapter so that it is highlighted, then click the Add button. A new network adapter should be added to your VM underneath the first network adapter we connected to the Bridged Network when we created this virtual machine. You should automatically be brought to this new network adapter to configure it. On the right side of the screen, on the drop-down menu labeled Virtual Switch:, select Management Network then click Apply. ||||||||||||||||||| ||||||||||||||||||| Click on Add Hardware again to highlight, select Network Adapter, then click the Add button again to create the third and final network card. This time we're going to select IPS 1 for the Virtual switch: drop-down. Click Apply once more. ||||||||||||||||||| ||||||||||||||||||| At this point, our pfSense VM should have three network cards attached to three different virtual switches. Next, click on SCSI Controller under the Hardware list on the left pane, and click the Remove button that appears. We're removing the SCSI controller because our VM has no use for it. Afterwards, click Apply. ||||||||||||||||||| ||||||||||||||||||| Scroll down on the left pane under Management, and click on Checkpoints. Under the section labeled Checkpoint Type, ensure the Enable checkpoints checkbox is checked, click on the Standard checkpoints radio button, then click Apply. ||||||||||||||||||| ||||||||||||||||||| Without going too deep into it, checkpoints are essentially HyperV's terminology for snapshots, or a way to restore the VM back to a particular configuration in an instant. We will be using standard checkpoints for all of our VMs from here on out. Note: If you are running Client Hyper-V on Windows 8.x, you may not have a choice between Production and Standard Checkpoints. You can disregard this section. Production checkpoints were added in with the Windows 10 version of Client Hyper-V. If you want to find out more about the differences between production and standard checkpoints, go here: ||||||||||||||||||| ||||||||||||||||||| v/manage/choosebetween-standard-or-production-checkpoints-in-hyper-v. 9.11 Installing pfSense With our changes in place, the VM is ready to be powered on to have the operating system installed. Power on the VM by right clicking it, and selecting Start from the menu. You should see the VM's State in the Hyper-V Manager console change from Off to Running. Right click on the VM again, and select Connect… to enable a console session to the virtual machine. After booting the VM and attaching to the virtual machine's console, select option 1, Boot Multi User [Enter] by hitting the enter key. Let pfSense boot up, and the system should automatically run the installer. Adjust your video, screenmap, and keymap settings as necessary, then select< Accept these Settings >. ||||||||||||||||||| ||||||||||||||||||| On the next screen, select < Quick/Easy Install >, and let pfSense perform its installation routines. The next screen will inform you that the install will erase the contents of the hard disk. Since our virtual disk is already empty, this doesn't matter in the least. Select OK, and let the installer run. ||||||||||||||||||| ||||||||||||||||||| Eventually, you will be asked whether the standard kernel or the embedded one should be used. Be sure to select< Standard Kernel >. Finally, the installer informs you to reboot the machine to boot from the hard drive. Allow the virtual machine to reboot in order to finish the installation procedure. After system has rebooted, right click on the VM in the Hyper-V Manager, and select Turn Off… 9.12 Final VM Settings ||||||||||||||||||| ||||||||||||||||||| There is one last configuration change we want to make to the pfSense VM's setting s before we continue. Make sure that the VM has been turned off and that the State reads as Off in the Hyper-V Manager. Next, right click on the VM in the Virtual Machines pane, and select Settings… to bring up the settings menu. On the left pane under Hardware, select DVD Drive. If you don't see it, double click on IDE Controller 1 to bring it up under the IDE controller, then click on it. On the DVD Drive screen, select the Remove button to uninstall the virtual DVD drive entirely, then click Apply. Before closing the settings menu for the pfSense VM, We need to record the MAC addresses for each of our network adapters. Under Hardware, on the left ||||||||||||||||||| ||||||||||||||||||| pane, look for the network adapter labeled Bridged Network. Next to the adapter, you'll notice a + sign. Either, click on the symbol, or double click on Network Adapter to display the Hardware Acceleration, and Advanced Features options. Select Advanced Features to display advanced settings for the network adapter. Under the section labeled MAC address, make sure to document or write down the MAC address displayed in the input boxes. Repeat this process for the Management Network and IPS 1 network adapters and record the MAC addresses for these interfaces. Note: In the illustration above, I opted to change the MAC address setting ||||||||||||||||||| ||||||||||||||||||| from Dynamic To Static. You are not required to do this. When you first boot a VM in Hyper-V, the hypervisor will automatically generate a MAC address for you that begins with 00-15-5D (the OUI for Microsoft). If you select Static, You can overwrite the MAC address that Hyper-V assigns to the interface, to one of your choosing. You can use this to define a MAC address with a custom OUI. You can use this to feature to fool some malware and/or applications that attempt to check the MAC address to identify whether or not a system is a VM or not. Keep this in mind for systems in which you plan on doing malware analysis. For the time being however, I recommend leaving the MAC address radio button set to Dynamic, and recording the MAC address that appears. 9.13 Network Configuration Now, we need to set up networking in the VM itself. In HyperV Manager, start the VM if it isn't already running, then connect to the console. Wait for the VM to finish booting. You'll be greeted by the Assign Interfaces wizard. The wizard will start by asking if you want to set up VLANS. Say no, because our virtual lab will NOT be using VLANs. Next, the wizard will ask you to define which interface should be assigned to what role. There should be three network interfaces displayed: ||||||||||||||||||| ||||||||||||||||||| When we performed the Initial VM Settings configuration, I had you record the MAC addresses for each of the network adapters, and document which vswitch they are connected to. - Bridged, Management, and IPS 1. Your goal right now is determine which network adapter in pfSense (hn0, hn1, andhn2) corresponds to the network adapter configuration in Hyper-V. For example, in the illustration above, hn0 with a MAC address of00:15:5d:01:11:1b has the MAC address of the network adapter connected to the Bridged Network. The1c MAC address ofhn1 corresponds to the network adapter attached to the Management Network and 1d MAC address ofhn2 corresponds to the network adapter attached to IPS 1. For our lab, the Bridged Network is our connection to the outside world, and therefore will serve as the WAN interface in pfSense. In my case, that means thathn0 is the WAN interface. The hn1 interface is connected to the Management Network and will serve as the LAN interface, and finally,hn2 connected to the IPS 1 network will serve as the OPT1 interface. ||||||||||||||||||| ||||||||||||||||||| The setup script will ask you to confirm your interface choices. If you are satisfied, type "y" to let pfSense configure the interfaces and initial system settings. When everything is done, you'll be dropped into a menu system. Note: If for some reason the Assign Interfaces wizard was not run automatically when you first booted your pfSense VM, or you made a mistake and you need to correct it, option 1 on the pfSense main menu, Assign Interfaces will run the wizard, and/or allow you to correct interface assignment errors as necessary. Now that we have assigned the network interfaces, we have to configure IP addresses for these interfaces. In most cases, the WAN interface will automatically get an IP address from the device on your physical network that provides DHCP services. If this is not happening, you may have to troubleshoot your physical network. This is beyond the scope of our guide. As for the LAN and OPT1 networks, we have to manually set the IP address, subnet mask, and DHCP scopes for these networks. Select option 2 in the pfSense main menu to get started. ||||||||||||||||||| ||||||||||||||||||| The configuration wizard starts by asking for the interface the settings should be changed for. We have to go through the process two times, for the LAN and the OPT1 interface individually. Here are the settings I recommend: LAN IP: 172.16.1.1 LAN Subnet bit count: 24 LAN upstream gateway address: LAN IPv6 address: Do you want to enable the DHCP server on LAN? y LAN DHCP start address: 172.16.1.10 LAN DHCP end address: 172.16.1.254 Do you want to revert to HTTP as the webConfigurator protocol? N OPT1 IP: 172.16.2.1 OPT1 Subnet bit count: 24 OPT1 upstream gateway address: OPT1 IPv6 address: Do you want to enable the DHCP server on OPT1? y OPT1 DHCP start address: 172.16.2.10 OPT1 DHCP end address: 172.16.2.254 Do you want to revert to HTTP as the webConfigurator protocol? n The reason we choose 1.10 and 2.10 as the DHCP start addresses is to reserve ||||||||||||||||||| ||||||||||||||||||| a few IP addresses for static DHCP allocations. Static DHCP allocations allow you to configure a DHCP server to always serve the same IP address when requested from a particular MAC address. We'll talk about this a little bit more later. For now, we should be done mucking around in the pfSense CLI. From here on out, we get to use the web UI to configure pfSense. Note: If you are using 172.16.1.0/24 or the 172.16.2.0/24 network ranges in your physical network, choose another network range to assign to the LAN and OPT1 interfaces to avoid network conflicts. Note: When you are configuring the LAN and OPT1 interface IP addresses, the configuration script will ask, Do you want to revert to HTTP as the webConfigurator protocol? Always say no to this. This allows the web UI to accept HTTP logins, and makes your firewall credentials vulnerable to sniffing over the network. Your pfSense main menu should look something like this when you're all done: If so, move on to setting up pfSense from the webConfigurator. Before doing so however, you must configure the vEthernet adapter attached to the Management Network on your Windows host. If you haven't already, you'll want to visit the Unbinding Network Protocols on Windows Virtual Adapters at a minimum. This section will guide you through configuring the vEthernet adapter attached to the Management Network with an IP address as well as unbinding network protocols on this adapter to increase the security of your ||||||||||||||||||| ||||||||||||||||||| hypervisor host. You may also want to consider reading Using Windows Firewall to Limit Exposure of Windows Hypervisor Hosts to further enhance the security of the Windows host when interacting with your lab VMs. 9.14 webConfigurator - Initial Setup On the Windows Client Hyper-V host, open your favorite web browser (I prefer Firefox and navigate to (or the IP address you assigned to the LAN interface of your pfSense system). The default credentials for access to the web interface are admin/pfsense. If this is your first time logging in, pfSense takes you through a nice little setup wizard. I'll highlight some of the important things to take note of, and/or change as necessary: Set the primary and secondary DNS servers you plan on using. I typically use 8.8.8.8 (Google public DNS) and 4.2.2.2 (Level 3 public DNS). If you're using this lab at work, your workplace may have restrictions and you may have to use their DNS servers instead. Uncheck the checkbox next to Block private networks from entering via WAN rule. Uncheck the checkbox next to Block Bogon Networks on this page as well. The other default settings on the first time setup wizard should be fine. Please note that you will be asked to change the password for the admin account as a part of the setup wizard. Document the password, and keep it somewhere safe. Our next order of business is to restrict access to the web interface of the firewall to the machine with the IP address 172.16.1.2. On the menu bar at the top of the page, select Firewall > Rules. Click on LAN to modify the firewall policy for the LAN interface. Click the Add button with the arrow facing up. ||||||||||||||||||| ||||||||||||||||||| This will add the firewall rule to the top of the firewall policy to where it will evaluated first. The Edit Firewall Rule page is fairly straightforward. I've highlighted the options to be aware of. ||||||||||||||||||| ||||||||||||||||||| When you are done, click the Save button at the bottom of the page. This will take you back to the previous page. Click on the Apply Changes button to save your new firewall rule. Next, we want to an alias. Navigate to Firewall > Aliases. On the Firewall Aliases page, click on IP, then click Add. Create an alias with the following settings: ||||||||||||||||||| ||||||||||||||||||| Click Save to be brought back to the previous page, then click Apply Changes. We just created an alias for RFC1918 networks (local networks that are not routable through the public internet) This will come in handy later for creating firewall rules around these networks. Our final action for now is to navigate to System > Advanced, and click to fill in the checkbox next to the option Disable webConfigurator anti-lockout rule, and click Save on the bottom of the page. Be aware that if you did not create the firewall rule above, this will lock you out of the webConfigurator as soon as you click Save. 9.15 Making Checkpoints Snapshot management is an important part of any virtual lab. In Hyper-V terminology, snapshots are known as checkpoints. Our next step will be to create a checkpoint for the pfSense VM, so that if we make a mistake in the configuration, you have a known good, somewhat functional baseline to fall back to. I'm going to show you how to create a snapshot for your pfSense VM. Before doing so, ensure that in the VM's settings that checkpoints are enabled, and that if you are running Client Hyper- V on Windows 10, that Standard checkpoints have been selected. In the Hyper-V Manager, under the Virtual Machines pane, ||||||||||||||||||| ||||||||||||||||||| right click on pfSense, and choose the Checkpoint option. After a moment or two, you'll notice under the Checkpoints pane you now have a new checkpoint available. Underneath that check point you should see a triangle and the word Now. If at any point you needed to revert back to this checkpoint, simply right click on the checkpoint you just made, and select Apply to restore the checkpoint. 9.16 pfSense Summary At the end of all this, you should have a pfSense VM with the following settings: ● 512MB of RAM ● 5GB of disk ● 1 Virtual CPU ● 3 Network interfaces: ○ 1 Bridged Network (WAN interface) ○ 1 Management Network (LAN interface) ○ 1 IPS 1 network (OPT1 interface) pfSense should be configured as followed: ● The WAN interface should be the Hyper-V network adapter connected to the Bridged Network. This interface should automatically get an IP address via DHCP. If this isn't happening, get some help with whoever administers your local network, or start troubleshooting. Alternatively, configure a static IP ||||||||||||||||||| ||||||||||||||||||| address ● The LAN interface should have an IP address of 172.16.1.1. ● The LAN network should be 172.16.1.0 with a /24 (255.255.255.0) subnet mask ● DHCP should be configured for this network with a scope of 172.16.1.10-172.16.1.254 ○ 172.16.1.3-172.16.1.9 are available for setting static DHCP mappings ● The OPT1 interface should have an IP address of 172.16.2.1 ● The OPT1 network should be 172.16.2.0 with a /24 (255.255.255.0) subnet mask ● DHCP should be configured for this network with a scope of 172.16.2.10-172.16.2.254 ○ 172.16.2.2-172.16.2.9 are available for static DHCP mappings ● 172.16.1.2 should have a firewall rule configured to allow it access to the firewall over HTTPS; this is the anti-lockout firewall rule ● You should have an alias for RFC1918 networks configured for use with the firewall policy. ● You should have at least one good checkpoint you can revert to if you need to redo a step or somehow got lost/confused 9.17 What's Next? Now that you have an operational pfSense VM, your next steps are to read the pfSense Firewall Rules and Network Services Guide. Make sure the firewall policies match those illustrated in the segmentation guide, and make sure that the pfSense VM is hosting all the services mentioned in the Core Network Services guide to make your firewall fully functional and ready to handle your lab network. 9.18 Final Connectivity Checks and Troubleshooting Once you have finished setting up your pfSense VM according to the instructions laid out, Open a console session to the pfSense VM, and attach your mouse and keyboard by clicking on the console. If the screen is blank, hit ||||||||||||||||||| ||||||||||||||||||| the Enter key to bring up the pfSense main menu. Select option 8) Shell to start a shell session on pfSense. The first, run the command ping -c 5 www.google.com The output from theping command should look similar to the output above. Next, run the commandnslookup google.com Again, you should have output similar to the illustration above. Finally, runcurl -I The output of your curl command should be similar to the screen capture above. If so, you may typeexit to leave the shell, and close your connection to the pfSense console. If you had any problems with any of the commands above, you've got troubleshooting to do. When it comes to troubleshooting, start at ||||||||||||||||||| ||||||||||||||||||| layer 1 of the OSI model (Physical) and work your way up. Here are some questions to help the troubleshooting process: ● Which of the commands above failed? Did they all fail? ● Have you checked your physical cabling or wireless network connectivity? ● Does your company/LAN have MAC filtering/port security on? You might need to modify switch settings for your drop, or talk to your network administrator. ● Have you checked your upstream gateway/SOHO router? Trying pinging the gateway's IP address to see if your pfSense VM can reach the gateway. Try running the traceroute command to google.com to see if and where your ping command is failing. ● Do you have an upstream firewall in place? Is 443/TCP allowed outbound? Is 53/UDP allowed outbound? Is ICMP allowed? ● Did you input the commands correctly? Try entering them again. No, I'm not patronizing you. Once you have verified that your firewall's connectivity, policies, and services are properly configured, SNAPSHOT THE VM BEFORE moving on and trying your hand at setting up the remaining VMs. 9.20 Your Turn The pfSense virtual machine required a lot of custom configuration, both in Hyper-V, as well as in the VM itself once the OS had been installed. Now that we have done that together, I'm going to have you create the rest of the virtual machines on your own -- with the exception of the Metasploitable 2 VM, since it's a special case for reasons you will see momentarily. Below are a set of spec sheets to help you create the remaining virtual machines for the lab on your own. Think of them as checklists you should be able to run through unaccompanied. All of our Linux-based virtual machines are built on Debian Linux based distros (both, Ubuntu and Kali Linux are derived from Debian), so setup and ||||||||||||||||||| ||||||||||||||||||| configuration of all of our lab VMs should be very similar. 9.21 Kali Linux VM Kali Linux is a popular penetration testing distro. Popular because hackers and script kiddies are lazy, and practically everything you need for performing a penetration test or joining anonymous is installed by default. Kali is going to be our loud, obnoxious attacker that we're going to use as a noise generator for the express purpose of generating events on our IPS VM, and to test the AFPACKET bridge's functionality. Perform the following tasks to create a Kali Linux VM: ● Download Kali Linux 64-bit here: ● Create a VM with the following settings: ○ Name the VM "Kali" ○ Make the VM Generation 1 ○ Allocate 4GB of RAM ■ Uncheck the Use Dynamic Memory for this virtual machine feature ○ Connect the VM's network adapter to the IPS 1 virtual switch ○ Allocate 80 GB of space for the vhdx file ○ Under Installation Options, select the Install from a bootable CD/DVDROM radio button, then select the Image file (.iso) option, and browse to the Kali Linux ISO you downloaded ● Once the VM is created, adjust the following settings: ○ 1-2 virtual CPUs (virtual CPUs can be adjusted under Settings > Hardware > Processor and adjust Number of Virtual Processors: to 2) ○ Remove the SCSI Controller ○ If you are running Windows 10, change the Checkpoints option (Under Settings > Management > Checkpoints) to Standard ● Install Kali Linux ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy. ||||||||||||||||||| ||||||||||||||||||| ● After the installation is completed, shut down, or turn off the VM and adjust the following settings: ○ Remove the DVD drive located under IDE Controller 1 ○ Under Network Adapter > Advanced Features in the MAC address section, document the MAC address of this network adapter. ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP, for the MAC address of adapter 1 for the Kali Linux VM; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to! ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command, in this case, will work from this virtual machine, due to the unique firewall policy on the OPT1 network. ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade ○ Create a snapshot for the VM 9.22 SIEM VM SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for log aggregator; we will be having our IP

|||||||||||||||||||| |||||||||||||||||||| Next, click on USB (OS X users should click on Ports, then click on the the USB tab, instead). Uncheck the Enable USB Controller checkbox to disable USB support for the VM. We are disabling the audio and USB controllers for our VM in order to reduce possible attack surface for virtual machine escapes, as well as ensure that host and VM are as isolated from one another as reasonably possible. |||||||||||||||||||| |||||||||||||||||||| Note: Virtual machine escapes or “VM escapes” are a specific type of exploit that an attacker will attempt to use in order to gain some sort of execution or access to resources on the hypervisor host itself. While VM escapes that result in code execution on the hypervisor host are rare, there are other means that attackers or malware could potentially abuse to attack the hypervisor host itself, such as through shared folders (file system folders that both the host and the VM can read/write to) or through guest extension software packages (special software packages installed on the VM itself that enable extra functionality, or access to host system resources). Therefore it is important to make your lab VMs as lean as reasonably possible in order to avoid compromising the hypervisor host system. Finally, click on Network. You will be greeted by a window pane that has multiple tabs. The first tab is labeled Adapter 1. First, check that the Enable Network Adapter checkbox is checked. On the Attached to: drop-down, make sure that Bridged Adapter is selected. Under the Name: dropdown, choose the physical network card you want to have virtual machine bridge to. Select the network interface you typically use on the host to connect to the internet. Click on theAdvanced arrow to display more network adapter options. One of the |||||||||||||||||||| |||||||||||||||||||| fields that gets displayed is labeled MAC Address:. Document the MAC address displayed. Click on the Adapter 2 tab, and enable the network adapter if it is disabled, just like with Adapter 1. Next, set Attached to: to Host-Only Adapter, and Name: should default to VirtualBox HostOnly Ethernet Adapter on Windows, orvboxnet0 on non-Windows hosts. Just like with Adapter 1, click on Advanced, and record the MAC address of this interface. |||||||||||||||||||| |||||||||||||||||||| Finally, click on the Adapter 3 tab. Enable the adapter, and set Attached To: to internal network. The default internal network name will be intnet on Windows, or intnet0 on non-Windows hosts. The default internal network name (in either case) is perfectly fine for this network adapter, just make sure you document the internal network name somewhere. Make sure to record the internal network name for this adapter for later reference. Be sure to click Advanced and document the MAC address of this network adapter as well. Click OK to exit the settings menu. |||||||||||||||||||| |||||||||||||||||||| 10.6 Installing pfSense Back at the main menu, click on the pfSense entry to highlight it, then click the big green start arrow to boot up the VM. A console window should appear that will allow you to interact with the VM as though you had a keyboard, mouse, and monitor directly attached to it. By default, if you click on the window, the virtual machine will take control of the mouse and keyboard. To detach the mouse and keyboard from the VM console, you have to enter a special key or keyboard combination called a hotkey. In Windows and Linux, this is the right CTRL key, while on OS X, this is the left Command key. Installing pfSense is very easy. Let the VM boot up, and the system should automatically run the installer. Adjust your video, screenmap, and keymap settings as necessary, then select < Accept these Settings >. |||||||||||||||||||| |||||||||||||||||||| On the next screen, select < Quick/Easy Install > and let pfSense do all the heavy lifting. The next screen will inform you that the install will erase the contents of the hard disk. Since our virtual disk is already empty, this doesn’t matter in the least. Select OK, and let the installer run. |||||||||||||||||||| |||||||||||||||||||| The installer will go through and install pfSense to the virtual hard disk. At a certain point you will be asked whether the standard kernel or the embedded one should be used. Make sure to select < Standard Kernel >. Finally, the installer informs you to reboot the machine to boot from the hard drive. The installation is done, however, we’re not going to reboot. Instead, shut the VM down. Click Machine, in the VM’s console window menu, and clicking ACPI Shutdown. 10.7 Final VM Settings We have to modify a few more settings of the VM before we start configuring pfSense properly. On the VirtualBox main menu, right click on pfSense and go back to the Settings… menu. Select System, then ensure the the Motherboard tab is selected. We need to modify the Boot Order options. First, uncheck the checkbox next to Optical, then make sure that Hard Disk is the only checked option remaining. Finally, highlight Hard Disk then click the up arrow to the right of the Boot Order list, until Hard Disk is at the top. This change ensures that the operating system installed on the hard drive is the first and ONLY bootable device for this VM. |||||||||||||||||||| |||||||||||||||||||| Next, click Storage. Under the Storage Tree pane, click on the icon that looks like a CD, then on the bottom of the pane, click on the icon with a minus sign on it. This will uninstall the virtual CD/DVD drive for this VM. This change ensures that the virtual hard drive is the only storage media available for this VM now. Click OK to exit the Settings menu. |||||||||||||||||||| |||||||||||||||||||| 10.8 Network Configuration On the VirtualBox main menu. Click on the pfSense VM to highlight it, then click the start arrow to boot the VM. Once the console appears, wait a moment for the VM to finish booting, and you should be greeted by the pfSense main menu. First things first, we have to correlate which interfaces in the VM correspond to adapter 1, adapter 2, and adapter 3 in the VirtualBox network settings. In the pfSense main menu, select option 8 to open up a shell. If you're not familiar with the command line interface in Linux/Unix, don’t worry; we won’t be here very long. Type the commandifconfig a | egrep “em|ether”. What this command does is run ifconfig -a, a command that shows all the information for all of the network interfaces installed that the OS recognizes. The pipe symbol “|” is taking the output from theifconfig command and “piping” it as input to theegrep command. We are telling egrep to process the output fromifconfigand ONLY show us lines that command “em” OR “ether”. If you did it correctly, the output should look something like this: |||||||||||||||||||| |||||||||||||||||||| In BSD/Linux terms, the MAC address is referred to as ether, shorthand for ethernet address. What we need to do is compare the ether addresses displayed here to figure out which interface in the VM (em0, em1, andem2) maps to which adapter (adapter 1, 2, and 3) in VirtualBox. This is why I had you document the MAC addresses for adapter 1, 2, and 3 earlier when we created the VM; We need to know which em interface corresponds to which adapter number in the VirtualBox network settings so we can make sure the interfaces in the VM are assigned and configured correctly. We need to know which interface in pfSense corresponds to which adapter so we can designate the WAN (Bridged network), LAN (Host-Only Network) and OPT1 (Internal Network) properly in pfSense. In my case, the ether address for em0 is00:00:27:dd:12:f8. This corresponds to the MAC address assigned to adapter 1 in VirtualBox (000027dd12f8-- exactly the same, without the colon separators), em1 corresponds to adapter 2, and em2 corresponds to adapter 3. Once you have figured out which interface corresponds to which VirtualBox adapter, type exit to leave the shell and go back to the pfSense main menu. Select option 1, Assign Interfaces to run the interface assignment wizard. To manage its network services and firewall policies, pfSense uses internal network interface descriptors (WAN, LAN, OPT1, etc.). Assigning the interfaces defines which of them (em0, em1, em2, etc.) corresponds to which of pfSense's internal descriptors. |||||||||||||||||||| |||||||||||||||||||| The WAN interface needs to be assigned to the Bridged network adapter (Adapter 1, the Bridged network). The LAN interface needs to be assigned to our Host-Only adapter (Adapter 2, the Management network), and finally, OPT1 needs to be assigned to our internal network adapter (Adapter 3, the IPS 1 network). In my case, em0 became the WAN interface,em1 the LAN interface, and em2 was mapped to the OPT1 interface. After confirming that the settings are correct, you should automatically be returned to the pfSense main menu. Now that we have assigned the network interfaces, we have to configure IP addresses for these interfaces. In most cases, the WAN interface will automatically get an IP address from the device on your physical network that provides DHCP services. If this is not happening, you may have to troubleshoot your physical network. This is beyond the scope of our guide. As for the LAN and OPT1 networks, we have to manually set the IP address, subnet mask, and DHCP scopes for these networks. Select option 2 in the pfSense main menu to get started. |||||||||||||||||||| |||||||||||||||||||| The configuration wizard starts by asking for the interface the settings should be changed for. We have to go through the process two times, for the LAN and the OPT1 interface individually. Here are the settings I recommend: LAN IP: 172.16.1.1 LAN Subnet bit count: 24 LAN upstream gateway address: LAN IPv6 address: Do you want to enable the DHCP server on LAN? y LAN DHCP start address: 172.16.1.10 LAN DHCP end address: 172.16.1.254 Do you want to revert to HTTP as the webConfigurator protocol? N OPT1 IP: 172.16.2.1 OPT1 Subnet bit count: 24 OPT1 upstream gateway address: OPT1 IPv6 address: Do you want to enable the DHCP server on OPT1? y OPT1 DHCP start address: 172.16.2.10 OPT1 DHCP end address: 172.16.2.254 Do you want to revert to HTTP as the webConfigurator protocol? n The reason we choose 1.10 and 2.10 as the DHCP start addresses is to reserve a few IP addresses for static DHCP allocations. Static DHCP allocations allow you to configure a DHCP server to always serve the same IP address when requested from a particular MAC address. We will talk about this a little bit more later. For now, we are done mucking around in the pfSense CLI. We will be using the webConfigurator to manage and configure our pfSense VM. Note: If you are using 172.16.1.0/24 or the 172.16.2.0/24 network ranges in |||||||||||||||||||| |||||||||||||||||||| your physical network, choose another network range to assign to the LAN and OPT1 interfaces to avoid network conflicts. Note : When you are configuring the LAN and OPT1 interface IP addresses, the configuration script will ask Do you want to revert to HTTP as the webConfigurator protocol? Always say no to this. This allows the web UI to accept HTTP logins, and makes your firewall credentials vulnerable to sniffing over the network. Your pfSense main menu should look something like this when you’re all done: Before moving on to setting up pfSense from the webConfigurator, you must configure the HostOnly network card attached to the hypervisor host. If you are using Windows as your host, and If you haven’t already, you’ll want to visit the Unbinding Network Protocols on Windows Virtual Adapters at a minimum. This will guide you through configuring the VirtualBox Host-Only Ethernet Adapter with an IP address as well as unbinding network protocols on this adapter to increase the security of your Windows hypervisor host. You may also want to consider Using Windows Firewall to Limit Exposure of Windows Hypervisor Hosts to further enhance the security of the Windows host when interacting with your lab VMs. Linux and OS X users can configure the IP address of their host-only network card by using the command utilitiesifconfig orip addr add. Try one of the following commands to set the IP address of the vboxnet0 interface: ifconfig vboxnet0 172.16.1.2 netmask 255.255.255.0 |||||||||||||||||||| |||||||||||||||||||| ip addr add 172.16.1.2/24 dev vboxnet0 Please note that you will need root permissions to run these commands, and that these settings will NOT persist between reboots of your hypervisor host system. Configuring IP address persistence for your host OS is beyond the scope of this guide. If you don’t want to mess around with ifconfig or theip command you can use the VirtualBox Preferences menu to set the IP address of vboxnet0 as well. Navigate to Network, select the Host-Only Networks tab, clickvboxnet0 to highlight it, then click the screwdriver icon. A small window will pop up. Select the Adapter tab if it isn’t already selected, then fill out the following: IPv4 Address: 172.16.1.2 IPv4 Network Mask: 255.255.255.0 Leave the rest of the fields alone, then click OK to exit the configuration menu, then click OK to exit preferences menu. 10.9 webConfigurator - Initial Setup On your VirtualBox host OS, open your favorite web browser (I prefer Firefox - and navigate to (or the IP address you assigned to the LAN interface of your pfSense system). The default credentials for access to the web interface are admin/pfsense. If this is your first time logging in, pfSense takes you through a nice little setup wizard. I’ll highlight some of the important things to take note of, and/or change as necessary: Set the primary and secondary DNS servers you plan on using. I typically use 8.8.8.8 (Google public DNS) and 4.2.2.2 (Level 3 public DNS). If you’re using this lab at work, your workplace may have restrictions and you may have |||||||||||||||||||| |||||||||||||||||||| to use their DNS servers instead. Uncheck the checkbox next to Block private networks from entering via WAN rule. Uncheck the checkbox next to Block Bogon Networks on this page as well. The other default settings on the first time setup wizard should be fine. Please note that you will be asked to change the password for the admin account as a part of the setup wizard. Document the password, and keep it somewhere safe. Our next order of business is to restrict access to the web interface of the firewall to the machine with the IP address 172.16.1.2. On the menu bar at the top of the page, select Firewall > Rules. Click on LAN to modify the firewall policy for the LAN interface. Click the Add button with the arrow facing up. This will add the firewall rule to the top of the firewall policy to where it will evaluated first. The Edit Firewall Rule page is fairly straightforward. I’ve highlighted the options to be aware of. |||||||||||||||||||| |||||||||||||||||||| When you are done, click the Save button at the bottom of the page. This will take you back to the previous page. Click on the Apply Changes button to save your new firewall rule. Next, we want to an alias. Navigate to Firewall > Aliases. On the Firewall Aliases page, click on IP, then click Add. Create an alias with the following settings: |||||||||||||||||||| |||||||||||||||||||| Click Save to be brought back to the previous page, then click Apply Changes. We just created an alias for RFC1918 networks (local networks that are not routable through the public internet) This will come in handy later for creating firewall rules around these networks. Our final action for now is to navigate to System > Advanced, and click to fill in the checkbox next to the option Disable webConfigurator anti-lockout rule, and click Save on the bottom of the page. Be aware that if you did not create the firewall rule above, this will lock you out of the webConfigurator as soon as you click Save. 10.10 Take a Snapshot Snapshotting is a VERY important concept with VMs. It lets you restore your virtual machines to its state in the past when the snapshot was taken.This allows you to undo misconfigurations, solve problems or potential issues that might affect the VM, in relatively easy fashion. We’re going to take a snapshot of pfSense in its current state. In the VirtualBox main menu, click on the icon that looks like a little camera in the upper right corner of the menu, labeled Snapshots. |||||||||||||||||||| |||||||||||||||||||| The view changes to display the snapshot manager. You will find another camera icon located in the upper left corner of right pane. Click on this icon to take a snapshot of the VM as it is right now. |||||||||||||||||||| |||||||||||||||||||| Clicking the camera icon brings up a window entitled Take Snapshot of Virtual Machine. It asks you to name the snapshot and give it a description. If you keep multiple snapshots make sure you keep good descriptions that include WHEN the snapshot was taken, and WHAT STATE the VM is in when you took the snapshot. Once you’ve named it and added a good, valid description, click OK to generate the snapshot. |||||||||||||||||||| |||||||||||||||||||| After a few seconds, the snapshot you just created shows up in the snapshot manager for the selected VM, and a little indicator appears next to the name of the VM, displaying the number snapshots available.Remember to delete old snapshots you are NOT planning to use any more, in order to conserve disk space. When you are done, you can click the cog labeled Details to return the VirtualBox main menu. |||||||||||||||||||| |||||||||||||||||||| If you ever have to restore a VM from a snapshot, first, shutdown/power off the VM, then navigate to the snapshot manager. Right click on the snapshot you wish to revert to. Choose Show Details to view the description you (hopefully) provided for the snapshot, or select Restore Snapshot to start the snapshot restore process. This will take a moment or two, and voila, your VM is restored to its previous state. Note that the icons above the snapshot pane perform the same function as those presented in the menu when right clicking the snapshot: the camera with the Uturn arrow performs a snapshot restore, the camera icon with the X over it deletes the highlighted snapshot, and the camera with the circle shows you details on the highlighted snapshot. The sheep icon allows you to make an exact copy (also known as a clone) of the VM in its current state or at any state it was in, during any of the snapshots. |||||||||||||||||||| |||||||||||||||||||| 10.11 pfSense Summary At the end of all this, you should have a pfSense VM with the following settings: ● 512MB of RAM ● 5GB of disk ● 1 Virtual CPU ● DVD drive disabled ● Audio disabled ● USB controller disabled ● 3 Network interfaces: ○ 1 Bridged (WAN interface) ○ 1 Host-Only (Management/LAN interface) ○ 1 Internal Network (IPS 1/OPT1 interface) pfSense should be configured as followed: ● The WAN interface should be the VirtualBox network adapter connected to the Bridged Adapter. This interface should automatically get an IP address via DHCP. If this isn’t |||||||||||||||||||| |||||||||||||||||||| happening, get some help with whoever administers your local network, or start troubleshooting. Alternatively, configure a static IP address ● The LAN interface should have an IP address of 172.16.1.1. ● The LAN network should be 172.16.1.0 with a /24 (255.255.255.0) subnet mask ● DHCP should be configured for this network with a scope of 172.16.1.10172.16.1.254 ○ 172.16.1.3-172.16.1.9 are available for setting static DHCP mappings ● The OPT1 interface should have an IP address of 172.16.2.1 ● The OPT1 network should be 172.16.2.0 with a /24 (255.255.255.0) subnet mask ● DHCP should be configured for this network with a scope of 172.16.2.10172.16.2.254 ○ 172.16.2.2-172.16.2.9 are available for static DHCP mappings. ● 172.16.1.2 should have a firewall rule configured to allow it access to the firewall over HTTPS; this is the anti-lockout firewall rule. ● You should have an alias for RFC1918 networks configured for use with the firewall policy. ● You should have at least one good snapshot you can revert to if you need to redo a step or somehow got lost/confused. 10.12 What’s Next? Now that you have an operational pfSense VM, your next steps are to read the pfSense Firewall Rules and Network Services Guide. Make sure the firewall policies match those illustrated in the segmentation guide, and make sure that the pfSense VM is hosting all the services mentioned in the Core Network Services guide to make your firewall fully functional and ready to handle your lab network. 10.13 Final Connectivity Checks and Troubleshooting Once you have finished setting up your pfSense VM according to the instructions laid out, Open a console session to the pfSense VM, and attach your mouse and keyboard by clicking on the console. If the screen is blank, hit the Enter key to bring up the pfSense main menu. Select option 8) Shell to start a shell session on pfSense. |||||||||||||||||||| |||||||||||||||||||| The first, run the command ping -c 5 www.google.com The output from the ping command should look similar to the output above. Next, run the commandnslookup google.com Again, you should have output similar to the illustration above. Finally, runcurl -I |||||||||||||||||||| |||||||||||||||||||| The output of your curl command should be similar to the screen capture above. If so, you may typeexit to leave the shell, and close your connection to the pfSense console. If you had any problems with any of the commands above, you’ve got troubleshooting to do. When it comes to troubleshooting, start at layer 1 of the OSI model (Physical) and work your way up. Here are some questions to help the troubleshooting process: ● Which of the commands above failed? Did they all fail? ● Have you checked your physical cabling or wireless network connectivity? ● Does your company/LAN have MAC filtering/port security on? You might need to modify switch settings for your drop, or talk to your network administrator. ● Have you checked your upstream gateway/SOHO router? Trying pinging the gateway’s IP address to see if your pfSense VM can reach the gateway. Try running the traceroute command to google.com to see if and where your ping command is failing. ● Do you have an upstream firewall in place? Is 443/TCP allowed outbound? Is 53/UDP allowed outbound? Is ICMP allowed? ● Did you input the commands correctly? Try entering them again. No, I’m not patronizing you. Once you have verified that your firewall’s c onnectivity, policies, and services are properly configured, SNAPSHOT THE VM BEFORE moving |||||||||||||||||||| |||||||||||||||||||| on and trying your hand at setting up the remaining VMs. 10.14 Your turn The pfSense virtual machine required a lot of custom configuration, both in VirtualBox as well as in the VM itself once the OS had been installed. Now that we have done that together, I’m going to have you create the rest of the virtual machines yourself -- with the exception of the Metasploitable 2 VM since it’s a special case for reasons you will see momentarily. Below are a set of spec sheets to help you create the remaining virtual machines for the lab on your own. Think of them as checklists you should be able to run through unaccompanied. All of our Linux-based virtual machines are built on Debian Linux based distros (both, Ubuntu and Kali Linux are derived from Debian), so setup and configuration of all of our lab VMs should be very similar. 10.15 Kali Linux VM Kali Linux is a popular penetration testing distro. Popular because hackers and script kiddies are lazy, and practically everything you need for performing a penetration test or joining anonymous is installed by default. Kali is going to be our loud, obnoxious attacker that we’re going to use as a noise generator for the express purpose of generating events on our IPS VM. I want you to perform the following tasks: ● Download Kali Linux 64-bit here: ● Create a VM with the following settings: ○ Name the VM “Kali” ○ Choose Linux as the type ○ Choose Debian (64-bit) as the version ○ 4GB of RAM ○ 80GB fixed size VDI ○ 1-2 virtual CPUs (virtual CPUs can be adjusted under VM Settings > System > Processor tab once the VM has been created) ● Once the VM is created, adjust the following settings: ○ Under the System settings, ensure that the VM is using a PS/2 Mouse ○ |||||||||||||||||||| |||||||||||||||||||| Under Storage, point the virtual cd/dvd drive to where you downloaded the Kali Linux ISO, so the VM has a bootable drive, and we have an actual OS to install ○ Disable Audio ○ Disable the USB controller ○ Under Network settings, make sure that Adapter 1 (and only Adapter 1) is enabled and that it is attached to Internal Network, and that the internal network is namedintnet (if hosting Virtualbox on Windows) orintnet0 (if hosting VirtualBox on a non-Windows OS). The internal network name should be the EXACT same as the internal network name for Adapter 3 on the pfSense VM ■ Document the MAC address for Adapter 1, under its advanced settings ● Install Kali Linux ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy ● After the installation is completed, power down the VM and adjust the following settings: ○ Remove the virtual optical drive under the Storage Tree in Storage settings ○ Under System settings, disable the Floppy and Optical drives, and make the Hard Disk the first device the VM boots from ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP for the MAC address of Adapter 1 for the Kali VM; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM’s IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter’s name is in the VM) was given the IP address you configured its static mapping for in pfSense ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the |||||||||||||||||||| |||||||||||||||||||| internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command, in this case, will work from this virtual machine, due to the unique firewall policy on the OPT1 network. ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade ○ Create a snapshot for the VM 10.16 SIEM VM SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for log aggregator; we will be having our IPS VM log its events here. We’re going to be running Splunk on this VM. Splunk is a commercial program for managing logs on a pretty large scale. By default, and with no licensing, Splunk only allows you to collect or index 500MB worth of logs per day, however, there are /ways/ around this that we’ll talk about later. Perform the following tasks to set up the SIEM VM: ● Download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: ● Create a VM with the following settings: ○ Name the VM “SIEM” ○ Choose Linux as the type ○ Choose Ubuntu (64-bit) as the version ○ 4GB of RAM ○ 80GB fixed size VDI ○ 1-2 virtual CPUs (virtual CPUs can be adjusted under VM Settings > System > Processor tab once the VM has been created) |||||||||||||||||||| |||||||||||||||||||| ● Once the VM is created, adjust the following settings: ○ Under the System settings, ensure the VM is using a PS/2 Mouse ○ Under Storage, point the virtual cd/dvd drive to where you downloaded the Ubuntu Server Linux ISO, so the VM has a bootable drive, and we have an actual OS to install ○ Disable Audio ○ Disable the USB controller ○ Under Network settings, make sure that Adapter 1(and only Adapter 1) is enabled and that it is attached to Host-Only Adapter, and that the name is VirtualBox Host-Only Ethernet Adapter -- the same as Adapter 2 on the pfSense VM ■ Document the MAC address for Adapter 1, under its advanced settings ● Install Ubuntu Server ○ Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You should not need to install any other roles or features for this server ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy ● After the installation is completed, power down the VM and adjust the following settings: ○ Remove the virtual optical drive under the Storage Tree in Storage settings ○ Under System settings, disable the Floppy and Optical drives, and make the Hard Disk the first device the VM boots from ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP, for the MAC address of Adapter 1 for the SIEM VM; assign it the IP address 172.16.1.3, making sure to enter a description that tells you which VM this static mapping belongs to ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM’s IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter’s name is in the VM) was given the IP address you configured its static mapping for in pfSense |||||||||||||||||||| |||||||||||||||||||| ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade ○ Create a snapshot for the VM 10.17 IPS VM This VM is going to be responsible for running the AFPACKET bridge between the IPS 1 and IPS 2 networks. In the case of VirtualBox, these are going to be two internal networks. Perform the following actions to set up the IPS VM: ● We will be installing Ubuntu Server 16.04.1 LTS, 64-bit. If you installed the SIEM VM already, you shouldn’t need to download the ISO again, but just incase, download it here: ● Create a VM with the following settings: ○ Name the VM “IPS” ○ Choose Linux as the type ○ Choose Ubuntu (64-bit) as the version ○ 2GB of RAM ○ 80GB fixed size VDI ○ 1-2 virtual CPUs (virtual CPUs can be adjusted under VM Settings > System > |||||||||||||||||||| |||||||||||||||||||| Processor tab once the VM has been created) ● Once the VM is created, adjust the following settings: ○ Under the System settings, ensure that the VM is using a PS/2 Mouse ○ Under Storage, point the virtual cd/dvd drive to where you downloaded the Ubuntu Server Linux ISO, so the VM has a bootable drive, and we have an actual OS to install ○ Disable Audio ○ Disable the USB controller ○ Under Network settings, make sure that Adapter 1, Adapter 2, and Adapter 3 are all enabled, and are using the following settings: ■ Adapter 1 should be attached to Host-Only Adapter named VirtualBox HostOnly Ethernet Adapter ● Under Adapter 1, be sure to click Advanced and document the MAC address of this interface; This is going to be the primary network interface for this host ■ Adapter 2 should be attached to Not attached ● Click on Advanced settings. Document the MAC address of this interface ■ Adapter 3 should be attached to Not attached ● Click on Advanced settings. Document the MAC address of this interface ● Install Ubuntu Server ○ The installer will reach a section titled Configure the network and ask you which network card is the primary network interface for this system ■ Usually, these interfaces are labeled eth0, eth1, and eth2, but may be labeled differently in your case ■ In almost all cases, the first network adapter added to a VM in VirtualBox will line up with the first network interface listed on the Configure the network screen (which is almost alwayseth0). Hit Enter, and wait for it to try and configure the interface via DHCP. ■ The installer will try to get an IP address for the network interface via DHCP. If this is not successful, select the option, and select one of the other two network interfaces listed, and try again. ■ If none of the interfaces work, verify the pfSense VM is running, connected to the Management Network, and that the DHCP service is enabled, then try again until DHCP configuration is successful. ○ Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You should not need to install any other roles or features for this server. ○ Note: If you configured the |||||||||||||||||||| |||||||||||||||||||| firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy. ● After the install is completed, power down the VM and adjust the following settings: ○ Remove the virtual CD/DVD under the Storage Tree in Storage settings ○ Under System settings, disable the Floppy and Optical drives, and make the Hard Disk the first device the VM boots from ○ Adapter 2 should be attached to Internal Network named intnet or intnet0 -the same as the kali VM ■ Under advanced settings, there is a dropdown labeled Promiscuous Mode:. Set this to Allow VMs (This is VERY important!) ○ Adapter 3 should be attached to Internal Network named intnet1. You will have to name this new internal network. This network is the IPS 2 network ■ Under advanced settings, there is a dropdown labeled Promiscuous Mode:. Set this to Allow VMs (This is VERY important!) ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP, for the MAC address of Adapter 1 for the IPS VM; assign the IP address 172.16.1.4, making sure to enter a description that tells you which VM this static mapping belongs to ● Power on the virtual machine and do the following: ○ Determine which interface in the IPS VM maps to which interface in the VM’s network settings ■ In the terminal, runifconfig -a | egrep HWaddr ● This command will show you the name of the of the interface and it’s HWaddr (Hardware Address - another name for MAC address). Compare this to the MAC addresses you documented earlier ● You need to determine the name of the interface that has the MAC address associated with Adapter 1 in the VirtualBox network settings. This is going to be the only network interface this VM can use to talk to other hosts over the network ○ Verify that the static mapping for the VM’s IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter’s name is in the VM) was given the IP address you configured its static mapping for in pfSense |||||||||||||||||||| |||||||||||||||||||| ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade ○ Create a snapshot for the VM 10.18 Promiscuous Mode In the configuration checklist above for the IPS VM, I very briefly mentioned configuring promiscuous mode for the network adapters on the. This setting MUST be set to Allow VMs. Promiscuous mode allows a network card that has this setting enabled to collect or sniff traffic observed on a given virtual network from other hosts. This configuration is extremely important and MUST be set on the network adapters connected to intnet/intnet0 and intnet1 for the IPS VM to perform its function correctly. |||||||||||||||||||| |||||||||||||||||||| 10.19 Metasploitable 2 Setting up Metasploitable 2 is a little bit different compared to the other virtual machines , because the VM comes in a prepackaged for use with VMware products. However, VirtualBox is able to utilize these prepackaged files as well. Download Metasploitable 2 from . Metasploitable 2 is distributed as a zip archive, so you need a compression utility to extract it. Linux/OS X/BSD users usually have the unzip command available. Open a console/shell, navigate to where the file was downloaded (typically~/Downloads), then run unzip metasploitable-linux-2.0.0.zip. You may want to move themetasploitable-linux2.0.0 directory to the same directory you are storing the VDI files for your other lab VMs. On Windows, I prefer 7-Zip (. Move the .zip file to the directory you are storing your VirtualBox VMs. In my case, I moved the .zip file toD:\vbox Right click on the zip file, and use 7-Zip to extract the files in the zip to metasploitablelinux-2.0.0 |||||||||||||||||||| |||||||||||||||||||| Fire up VirtualBox and select New to create a new VM. Name the VM “Metasploitable 2”. For the OS type, select Linux. For the version, select Ubuntu (64-bit). Then click Next. Allocate 512MB of RAM to the VM, and click Next. On the next screen, click the Use an existing virtual hard disk file radio button, and click the folder with the ^ symbol over it. This will open an |||||||||||||||||||| |||||||||||||||||||| explorer window. Navigate to the metasploitable-linux-2.0.0 directory we extracted from the Metasploitable 2 zip file moments ago. Inside, is a subdirectory named Metasploitable2-Linux. Double click to enter that directory. In my case, the full directory path is D:\vbox\metasploitable-linux2.0.0\Metasploitable2-Linux. Double click on theMetasploitable.vmdk file. Selecting the vmdk file should have brought you back to the Hard disk screen. Click the create button. The virtual machine is added to the main VirtualBox menu. |||||||||||||||||||| |||||||||||||||||||| The Metasploitable 2 VM should be added to our list on the VirtualBox main menu. Next up is something you should be familiar with by now: we are going to adjust the settings of the virtual machine. ● Go to the System screen. Under Boot Order, uncheck the Floppy and Optical options, and make the Hard Drive the first boot option. Change the Pointing Device: to PS/2 Mouse ● Remove the virtual CD/DVD under the Storage Tree in Storage settings ● Disable Audio ● Disable the USB controller ● Navigate to Network and under the Adapter 1 tab, ensure the adapter is enabled. Attach the adapter to Internal Network named intnet1 (the EXACT same as Adapter 3 for the IPS VM). Click on Advanced and document the MAC address of the interface. ● In pfSense, under Services > DHCP Server, add a static DHCP mapping under the |||||||||||||||||||| |||||||||||||||||||| OPT1 interface for this virtual machine. Give it the IP address 172.16.2.3 and make sure to enter a description that states which VM the mapping belongs to! At this point, the VM should be bootable. Feel free to power it on to make sure there aren’t any problems. If you’re greeted with a login prompt, then this indicates everything is working fine. Power down the virtual machine and take a snapshot. Note: This VM is located on intnet1, which we haven’t bridged to intnet/intnet0 with the IPS VM yet. So in spite of having a static DHCP mapping, it has no way of reach out to the pfSense virtual machine for an IP address yet. We are going to fix this soon. 10.20 Next Steps All of the VirtualBox setup steps have been completed. However, you’re not quite done yet. Here is a checklist of tasks to complete: ● If you are running Virtualbox on a Windows host, and haven’t completed the chapter Defense in Depth for Windows Hosted Hypervisors, I would very highly suggest doing so in order to harden your hypervisor host. ● While not strictly necessary, you may also want to complete the Remote Lab Management guide. Specifically, the sections related to Windows Remote Access, Enabling SSH on Kali Linux, and if you’re lazy like me, the section on Securing root SSH access. This will allow you to remotely manage all of your lab VMs, as well as finish the IPS and Splunk setup guides much more easily |||||||||||||||||||| |||||||||||||||||||| than through the VirtualBox console. ● You still need to install IDS/IPS software on the IPS VM to get a functioning AFPACKET bridge between the IPS 1 and IPS 2 networks. Check out the IPS Installation Guide to learn how to do this with either Snort or Suricata as your IPS software of choice. ● The SIEM VM needs to have Splunk installed and configured. You’ll need to complete the Splunk Installation Guide. ● You may want to consider reading the Automated Patching for Linux Lab VMs chapter, and implementing theupdater.sh script for your Linux VMs. ● Do you want some ideas on where to take your lab? Check out the chapter In Your Own Image, for some tips on how to mold your VM lab to better suit your needs. 11 Setup - VMware Fusion Pro Note: This guide is written for VMware Fusion Pro, specifically, the latest version as of writing, version 8.5. Fortunately for us, OS X is the only OS VMware Fusion Pro was designed for is OS X. This guide was written using OS X 10.12 Sierra, the latest version as of writing, but should work just fine with 10.11 El Capitan. Be aware that this guide specifically requires VMware Fusion Pro. Pro edition allows us to modify virtual networking settings easily and create additional virtual networks through its virtual network editor tool. We need the ability to modify and create virtual networks easily, and only Fusion Pro has a network editor application. 11.1 Installation First, you need to download VMware Fusion Pro from VMware website. You can a trial copy here: . Run the installer; the default settings should be fine. If you have a license, enter it when prompted to do so during the installation. If you do not have a license, continue on and use the trial time you have for now. |||||||||||||||||||| |||||||||||||||||||| 11.2 Hypervisor Preferences Start up VMware Fusion Pro and you’ll be greeted by a screen that automatically assumes you want to create or add a new VM. Before we go down that route however, we have to make some changes to the hypervisor. In the main menu bar, click on VMware Fusion, then click on Preferences… . This will open the hypervisor preferences menu. We’re interested in the Network options, so click on the Network icon. |||||||||||||||||||| |||||||||||||||||||| The Network menu allows you to define network bridging settings, as well as define custom network settings. Before you can change anything, however, you need to click the lock on the bottom of the menu, and input your password, when prompted. After doing so, uncheck the Require authentication to enter promiscuous mode checkbox. By doing so, you will no longer be prompted to enter your Mac user's password every time a VM's network interface switches to promiscuous mode to perform certain tasks. As an example, the IPS VM we are going to set up needs promiscuous mode to be enabled.

where total number of cores is no more than 2 ○ Connect the Network Adapter tovmnet3 ■ Ensure that Connect Network Adapter checkbox is checked ■ Under Advanced options, click the Generate button to generate a new MAC address. Record this MAC address as well as the VM it corresponds to. ○ Remove the USB Controller ○ Remove the Sound Card ○ Remove the Printer ○ Remove the Camera ○ In the Isolation settings, uncheck the Enable Drag and Drop and Enable Copy and Paste ● Install Kali Linux ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy ● After the installation is completed, power off the VM and adjust the following settings: ○ Remove the virtual CD/DVD drive ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP for the MAC address of the Kali VM's network adapter; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM's IP address worked; open a |||||||||||||||||||| |||||||||||||||||||| terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command, in this case, will work from this virtual machine, due to the unique firewall policy on the OPT1 network ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade 11.14 SIEM VM SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for log aggregator; we will be having our IPS VM log its events here. We're going to be running Splunk on this VM. Splunk is a commercial program for managing logs on a pretty large scale. By default, and with no licensing, Splunk only allows you to collect or index 500MB worth of logs per day, however, there are /ways/ around this that we'll talk about later. I want you to perform the following tasks to set up the SIEM VM: ● Download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: ● Create a new, custom virtual machine with the following settings: ○ Configure the VM to install from the Ubuntu Linux ISO you downloaded ○ Set the Guest OS family to Linux |||||||||||||||||||| |||||||||||||||||||| ○ Set the Guest OS version to Ubuntu 64-bit ○ Create a new virtual disk ○ Name the VM "SIEM" ○ Note that after saving the VM and granting it a name, VMware Fusion immediately assumes that you want to turn the virtual machine on. In order to adjust the hardware settings as described below, go to the main menu, click Virtual Machine, then select Shut Down. ● Change the following hardware settings (Virtual Machine > Settings or you can click the Customize Settings button on the Finish page of the new VM wizard): ○ Under CD/DVD (IDE), select the Choose a disc or disc image option, and find the Ubuntu 16.04.1 LTS 64-bit Linux ISO you previously downloaded ■ Ensure that the Connect CD/DVD Drive option is checked ○ Allocate 4GB (4096MB) of RAM ○ Modify the Hard Disk (SCSI) settings ■ Change the disk size to 80.00GB ■ Uncheck the Advanced option Split into multiple files ■ Check the Advanced option Pre-allocate disk space ○ Allocate 1-2 processors, where total number of cores is no more than 2 ○ Connect the Network Adapter tovmnet2 ■ Ensure that Connect Network Adapter checkbox is checked ■ Under Advanced settings, click the Generate button to generate a new MAC address. Record this MAC address as well as the VM and network it corresponds to. ○ Remove the USB Controller ○ Remove the Sound Card ○ Remove the Printer ○ Remove the Camera ○ In the Isolation settings, uncheck the Enable Drag and Drop and Enable Copy and Paste ● Install Ubuntu Server ○ Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You should not need to install any other roles or features for this |||||||||||||||||||| |||||||||||||||||||| server. ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy. ● After the installation is completed, power off the VM and adjust the following settings: ○ Remove the virtual CD/DVD drive ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP, for the MAC address of the SIEM VM's network adapter; assign it the IP address 172.16.1.3, making sure to enter a description that tells you which VM this static mapping belongs to! ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade 11.15 IPS VM |||||||||||||||||||| |||||||||||||||||||| This VM is going to be responsible for running the AFPACKET bridge between the IPS 1 (vmnet3) and IPS 2 (vmnet4) virtual networks. Perform the following tasks to install the IPS VM: ● Download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: ● Create a new, custom virtual machine with the following settings: ○ Configure the VM to install from the Ubuntu Linux ISO you downloaded ○ Set the Guest OS family to Linux ○ Set the Guest OS version to Ubuntu 64-bit ○ Create a new virtual disk ○ Name the VM "IPS" ○ Note that after saving the VM and granting it a name, VMware Fusion immediately assumes that you want to turn the virtual machine on. In order to adjust the hardware settings as described below, go to the main menu, click Virtual Machine, then select Shut Down. ● Change the following hardware settings (Virtual Machine > Settings or you can click the Customize Settings button on the Finish page of the new VM wizard): ○ Under CD/DVD (IDE), select the Choose a disc or disc image option, and find the Ubuntu 16.04.1 LTS 64-bit Linux ISO you previously downloaded ■ Ensure that the Connect CD/DVD Drive option is checked ○ Allocate 2GB (2048MB) of RAM ○ Modify the Hard Disk (SCSI) settings ■ Change the disk size to 80.00GB ■ Uncheck the Advanced option Split into multiple files ■ Check the Advanced option Pre-allocate disk space ○ Allocate 1-2 processors, where total number of cores is no more than 2 ○ Connect the Network Adapter tovmnet2 ■ Ensure that Connect Network Adapter checkbox is checked ■ Under Advanced settings, click the Generate button to generate a new MAC address. Record this MAC address as well as the VM and network it corresponds |||||||||||||||||||| |||||||||||||||||||| to. ○ Remove the USB Controller ○ Remove the Sound Card ○ Remove the Printer ○ Remove the Camera ○ In the Isolation settings, uncheck the Enable Drag and Drop and Enable Copy and Paste ○ Add two more network adapters ■ Attach Network Adapter 2 tovmnet3 (IPS 1) ● Under Advanced settings, click the Generate button to generate a new MAC address. Record this MAC address as well as the VM and network it corresponds to. ● Uncheck the Connect Network Adapter checkbox! ■ Attach Network Adapter 3 tovmnet4 (IPS 2) ● Under Advanced settings, click the Generate button to generate a new MAC address. Record this MAC address as well as the VM and network it corresponds to. ● Uncheck the Connect Network Adapter checkbox! ● Install Ubuntu Server ○ The installer will reach a section titled Configure the network and ask you which network card is the primary network interface for this system ■ Usually, these interfaces are labeled eth0, eth1, and eth2, but may be labeled differently in your case ■ In almost all cases, the first network adapter added to a VM in VMware Fusion will line up with the first network interface listed on the Configure the network screen (which is almost alwayseth0). Hit Enter, and wait for it to try and configure the interface via DHCP. ■ The installer will try to get an IP address for the network interface via DHCP. If this is not successful, select the option, and select one of the other two network interfaces listed, and try again. ■ If none of the interfaces work, verify the pfSense VM is running, connected to the Management network, and that the DHCP service is enabled, then try again until DHCP configuration is successful. ○ Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You should not need to |||||||||||||||||||| |||||||||||||||||||| install any other roles or features for this server. ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy. ● After the install is completed, power off the VM and adjust the following settings: ○ Remove the virtual CD/DVD drive ○ Navigate to Network Adapter 2, and check the Connect Network Adapter checkbox ■ Confirm Network Adapter 2 is connected to vmnet3 ○ Navigate to Network Adapter 3, and check the Connect Network Adapter checkbox ■ Confirm Network Adapter 3 is connected tovmnet4 ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP, for the MAC address of the original network adapter for the IPS VM (attached to the Management network,vmnet2); assign it the IP address 172.16.1.4, making sure to enter a description that tells you which VM this static mapping belongs to! ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense. ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail |||||||||||||||||||| |||||||||||||||||||| ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade ○ Create a snapshot for the VM 11.16 Metasploitable 2 Metasploitable 2 is a little bit different from the other VMs. As it is already pre-created, all we need to do is import it, and reconfigure some of the virtual hardware. This should be pretty simple to do. First, download a copy of Metasploitable 2 from . Metasploitable 2 is distributed as a zip archive. The Finder application will unzip the file if you double click on it; alternatively you can use the command line utilityunzip to do so. After downloading and decompressing the archive, you will be left with a folder labeled metasploitable2-linux-2.0.0.0 in your Downloads directory. Inside of that directory is another directory named Metasploitable2-Linux. Move the entireMetasploitable2-Linux folder into the directory where the rest of your VMware virtual machines' folders are stored; in my case, this is/Users/[username]/Documents/Virtual Machines/. In VMware Fusion, click Open... from the File option in the menu bar. A Finder window appears in which you can. navigate to the folder/Users/[username]/Documents/Virtual Machines/Metasploitable2-Linux. By double clicking on the file labeled Metasploitable.vmx, the Metasploitable VM gets registered with VMware Fusion, and becomes ready to be run. |||||||||||||||||||| |||||||||||||||||||| Before powering it on though, we have to adjust some of its settings. Click on the wrench in the Metasploitable2-Linux console window or use Virtual Machine > Settings... to bring up the Settings menu for Metasploitable 2. Perform the following actions: ● Reconfigure the Network Adapter and connect it tovmnet4(IPS 2) ○ Under Advanced options, record the MAC address of this adapter, or generate a new one via the Generate button. Record this MAC address as well as the VM it corresponds to. ● Remove Network Adapter 2 ● Remove the USB Controller ● Remove the CD/DVD drive ● In the Isolation settings, uncheck the Enable Drag and Drop and Enable Copy and Paste ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP, for the MAC address of the Network Adapter for the Metasploitable 2 VM (attached to the IPS 2 network,vmnet4); assign it the IP address 172.16.2.3, making sure to enter a description that tells you which VM this static mapping belongs to! |||||||||||||||||||| |||||||||||||||||||| Power on the Metasploitable 2 VM, make sure it is bootable. If so, take a snapshot of the virtual machine, it's ready to go for now. You won't be able to verify whether the DHCP mapping works until AFTER the IPS VM is configured. 11.17 Next Steps The VMware Fusion initial setup is all but done at this point. However, you're not quite out of the woods just yet. Here is a checklist of tasks to complete: ● While not strictly necessary, you may also want to complete the Remote Lab Management guide. Specifically, the sections related to Linux, BSD and OSX Remote Access, Enabling SSH on Kali Linux, and if you're lazy like me, the section on Securing root SSH access. This will allow you to remotely manage all of your lab VMs, as well as finish the IPS and Splunk setup guides much more easily than through the VirtualBox console. ● You still need to install IDS/IPS software on the IPS VM to get a functioning AFPACKET bridge between the IPS 1 and IPS 2 networks. Check out the IPS Installation Guide to learn how to do this with either Snort or Suricata as your IPS software of choice. ● The SIEM VM needs to have Splunk installed and configured. You'll need to complete the Splunk Installation Guide. ● You may want to consider reading the Automated Patching for Linux Lab VMs chapter, and implementing theupdater.sh script for your Linux VMs. ● Do you want some ideas on where to take your lab? Check out the chapter In Your Own Image, for some tips on how to mold your VM lab to better suit your needs. 12 Setup - VMware Workstation Pro Note: This guide is written for VMware Workstation Pro, specifically, the latest version as of this writing, version 12.5. I'll be writing this guide using Windows as the host operating system, though the host OS itself should not matter too much; you should be able to easily follow along using Windows or |||||||||||||||||||| |||||||||||||||||||| Linux as the host OS. VMware Workstation is a hosted hypervisor offered by VMware. Our guide specifically needs the Pro edition of workstation due to enhanced virtual networking features that aren't available in the standard edition of VMware Workstation. 12.1 Installation Installing VMware Workstation is pretty straightforward. You can get a trial copy to download and use for free, without even having an account with VMware. As of right now, this is the link to VMware Workstation (Windows and Linux can be downloaded on this page): Run the installer and go through the installation steps; the whole process should finish quickly. Please note that when installing on Linux, you'll need to install kernel modules for VMware as a part of the software installation. To do this, usually you'll need the headers for the kernel you're currently running available. While the name of the package(s) that include the headers varies from distro to distro, and from package manager to package manager, usually the headers are included in some sort of a kernel development package. Consult documentation for your Linux distro on how to acquire the kernel headers. 12.2 Hypervisor Preferences Now that you have VMware Workstation installed, start it up. We're going to make some configuration changes here and there, in order to support our lab environment. On first start, the program asks you if you want to input a license key or use the trial period. If you opt to use the trial period, you have to provide a valid e-mail address for VMware to send spam to your inbox. Well, they asked for a valid one, which doesn't necessarily mean your e-mail address. I suggest you to have a look at 10 Minute Mail, Guerilla Mail or a similar services. I used 10 Minute Mail for my trial period with no problems to report. After entering your license key or e-mail address, you're greeted |||||||||||||||||||| |||||||||||||||||||| with a main menu that looks like this: Click Edit on the menu bar, then click Preferences. A new window appears where you can modify a variety of settings for VMware Workstation. You may be interested in checking the other settings and setting things to your liking, but for the time being, all we're concerned with is the option titled Workspace, specifically, the setting related to Default location for virtual machines. By default, virtual machine files and folders will be placed under your home directory. If you want to change where they are stored, click the Browse button, and choose a new folder to store your VM files in. In my case, I chose to store my virtual machines inD:\WorkstationVMs. When you are done configuring this, or if you are satisfied with the defaults, click OK to close the Preferences window. 12.3 Virtual Networks Back in the main menu, choose Edit from the menu bar, then click on Virtual Network Editor to open the corresponding window. |||||||||||||||||||| |||||||||||||||||||| By default, VMware Workstation has three virtual networks defined VMnet1, a host-only network, and VMnet8, a NAT network, and VMnet0, a network for bridging to the host system's network connection. For our lab,VMnet0 will serve as our Bridged Network, VMnet1 will be our Management Network. Additionally, we will be creating two more host-only virtual networks to server as the IPS 1 Network and the IPS 2 Network, respectively. Note: If you are using VMware Workstation on Windows, the VMnet0 network will not be visible, and you will NOT be able create additional virtual networks until you click the Change Settings button in the lower right corner of the Virtual Network Editor window. Afterwards, the greyed out options should be available to edit, and VMnet0 should be visible. Linux users should not have to worry about this. |||||||||||||||||||| |||||||||||||||||||| Click to select VMnet1 in the main window pane, then uncheck the checkbox Use local DHCP service to distribute IP address to VMs in theVMnet Information section. Next, we are going to create two additional host-only networks. To start, click on the Add Network... button underneath the main window pane. A small window pops up with a drop-down labeled Select a network to add:. Click OK to choose the default network name (in my case, this was VMnet2). This network is going to serve as our lab's IPS 1 Network. |||||||||||||||||||| |||||||||||||||||||| After a moment or two, you should be back at the Virtual Network Editor. The window should have highlightedVMnet2 in the primary pane for you already; if it did not, click to selectVMnet2. In theVMnet Information section, verify that the radio button in front of the Host-only (connect VMs internally in a private network) option is selected; this should be the default setting. Uncheck the option Connect a host virtual adapter to this network and the option Use local DHCP service to distribute IP address to VMs. |||||||||||||||||||| |||||||||||||||||||| You might be wondering why we're disabling DHCP and network adapters for VMnet1 and VMnet2so far, and why we've disabled the host virtual adapter for VMnet2. The reason why we did all that is that we are going to configure pfSense to handle DHCP services, as opposed to VMware Workstation. Leaving DHCP enabled on the VMware virtual networks would cause IP address conflicts; beyond that pfSense's DHCP server offers much more flexibility. Regarding the removal of the host virtual adapter, we're doing this because VMnet2 will be one of the two IPS networks. As we want to protect the hypervisor host from potential threats, we do not want the host OS to have direct access to theVMnet2 network. We still need an IPS 2 Network, which we are going to create by repeating the process we went through above. Add a new network (in my case the network name was VMnet3), verify that it's type is Host-only, ensure that the checkboxes Connect a host virtual adapter to this network, and Use local DHCP service to distribute IP address to VMs, are unchecked. After completing this task, the primary pane of the Virtual Network Editor should look like this: The virtual networks are now configured to support our lab environment. Click the Apply button, then click OK to confirm the adjusted network settings, and close the Virtual Network Editor window. Creating the First VM, pfSense pfSense is the keystone holding this entire configuration together. Personally, it's my favorite firewall distro due to ease of use, the amount of functionality it includes out of the box, combined with a plugin/add-on system for additional functionality. If you have the CPU, RAM, and disk, pfSense can easily be |||||||||||||||||||| |||||||||||||||||||| converted into a socalled "Next-Generation" firewall. Make your way to . Download the latest installation ISO for the amd64 architecture. While you're at it, you may want to download a compression utility. The pfSense maintainers distribute pfSense as an ISO image file compressed with gzip. This means that we'll need to decompress the ISO file at some point. On Windows, I prefer 7-Zip (as my compression utility of choice for decompressing files, since 7-Zip can handle zip, gzip, rar, and 7z files (among others) easily. Linux on the other hand, usually includes the gzip orgunzip command line utilities for decompressing gzipped files. 12.4 Adding a New VM In the VMware Workstation main window, click File on the menu bar, and select New Virtual Machine to start the New Virtual Machine Wizard. The first screen of the wizard asks you to choose the type of configuration you would like to proceed with. Select the radio button next to Typical (recommended), then click Next. The custom (advanced) configuration type allows you to configure a bunch of additional settings on your VM relating to disk types and storage options. As none of them need to be adjusted for the purposes of our lab, creating virtual machines with the suggested typical configuration will serve us just fine. |||||||||||||||||||| |||||||||||||||||||| The next screen allows us conf igure if and how we want to install our VM's operating system. Select the radio button next to Installer disc image file (iso): radio button, then click Browse, and an explorer window opens. Navigate to where you stored the compressed pfSense ISO file, and if you haven't already, decompress it as necessary. In my case, the ISO inD:\ISOs\Nix. Note that you'll get a warning from the wizard, stating,Could not detect which operating system is in this disc image. Do not be concerned, VMware workstation attempts to make educated guesses as to what type and version of operating system you are attempting to load. We will be manually specifying the operating system and version shortly.. After selecting the ISO file, click Next. |||||||||||||||||||| |||||||||||||||||||| On the next screen, the wizard asks you to select the guest operating system and its version you are planning to install, since it failed to detect the OS on the previous step.. Click the radio button next to Other in the Guest operating system section, then select FreeBSD 64-bit from the Version drop-down list. To go ahead, click Next. |||||||||||||||||||| |||||||||||||||||||| The next screen asks you to name the virtual machine and to confirm the location where the corresponding files will be stored. Enter "pfSense" as the name of the VM, then click Next. |||||||||||||||||||| |||||||||||||||||||| On the next screen, you are asked to specify how much space you want to allocate for your VM's virtual hard disk. Enter the value "5.0" into the Maximum disk size (GB): input box, then select the radio button in front of Store virtual disk as a single file. Click Next to proceed. |||||||||||||||||||| |||||||||||||||||||| The final screen displays an overview of the current settings for this VM. As we have to adjust parts of the hardware present on the virtual machine, click the Customize Hardware... button. |||||||||||||||||||| |||||||||||||||||||| A new window appears, listing the hardware components currently attached to our new VM. In the Device pane on the left, click to highlight the Memory entry to bring up the related information and settings in the pane to the right. We're going to double the default amount of 256MB to 512MB, by inputting "512" into the Memory for this virtual machine: input box. |||||||||||||||||||| |||||||||||||||||||| Next, click on Network Adapter in the Device pane to bring up the settings for this network adapter. Ensure that the option Connect at power on is checked in the Device Status section. In the area below, titled Network connection, change the radio button from the default setting of NAT: Used to share the host's IP address to Bridged: Connected directly to the physical network. |||||||||||||||||||| |||||||||||||||||||| Afterwards, click the Add... at the bottom of the Device pane to start the Add Hardware Wizard. Click Network Adapter under the pane labeled Hardware types:, then click Next. |||||||||||||||||||| |||||||||||||||||||| On the next screen, Click the radio button next to Custom: Specific virtual network, then select VMnet1 in the drop-down menu. Make sure that the Connect at power on checkbox is checked, then click Finish. |||||||||||||||||||| |||||||||||||||||||| Repeat this process once more, creating a third network adapter, except this time, connect it to theVMnet2 virtual network. After adding the two network cards (for a grand total of 3 network adapters attached to this VM), we're going to remove the USB Controller and the Sound Card. In the Device list of the Hardware window, click on the USB Controller to highlight it, then click the Remove button underneath the device list. |||||||||||||||||||| |||||||||||||||||||| Repeat this process for the Sound Card to remove that device as well. After you are finished, your hardware configuration should look like this: Click the Close button in the lower right corner of the Hardware window to exit. Back in the New Virtual Machine Wizard, click Finish to let VMware |||||||||||||||||||| |||||||||||||||||||| Workstation create the VM. The system will take a moment or two to allocate the disk space to the virtual machine. When it is finished, the main window of VMware Workstation should list the pfSense VM under the Library pane. 12.5 Installing pfSense At this point, you're now ready to install pfSense on your VM. To start the virtual machine, in the VMware Workstation main menu, click on pfSense in the Library pane, then select Power on this virtual machine in the main window. After doing so, the primary window will switch to view the console of the powered on VM. Clicking on this console will attach your mouse and keyboard to this virtual machine; to detach the devices again, press ctrl+alt. . Let pfSense boot up, and the system should automatically run the installer. Adjust your video, screenmap and keymap settings as necessary, then select< Accept these Settings >. On the next screen, select < Quick/Easy Install > and let pfSense do all the heavy lifting. The next screen will inform you that the install will erase the contents of the hard disk. Since our virtual disk is already empty, this doesn't |||||||||||||||||||| |||||||||||||||||||| matter in the least. Select OK, and let the installer run. The installer will go through and install pfSense to the virtual hard disk. At a certain point you will be asked whether the standard kernel or the embedded one should be used. Make sure to select < Standard Kernel >. Finally, the installer informs you to reboot the machine to boot from the hard drive. The installation is done, however, we're not going to reboot. Right click on pfSense in the Library pane, select Power, and Power Off to power off the VM. |||||||||||||||||||| |||||||||||||||||||| 12.6 Final VM Settings Now that pfSense is installed, we no longer need the CD/DVD drive attached to the VM, so we're going to remove it. On the VMware Workstation main screen, right click on pfSense in the Library pane, then select Settings... In the Hardware tab, in the Devices pane, select CD/DVD (IDE), then click on the Remove button to remove the CD/DVD drive, just like we did with the |||||||||||||||||||| |||||||||||||||||||| sound card and the USB controller earlier. Next, select Network Adapter in the Device pane, then click on the Advanced... button to bring up the Network Adapter Advanced Settings menu. Document the contents of the MAC Address section of this screen; click OK when you are done. Repeat this process for Network Adapter 2 and Network Adapter 3. The goal is for you to know which MAC address corresponds to which VMnet (e.g. Bridged, VMnet1 (Management), andVMnet2 (IPS 1)) as we need this information to properly configure the network settings in pfSense later. Click OK to exit the Virtual Machine Settings menu. Your VM's Device listing should look like this: 12.7 Network Configuration Power on the pfSense VM, and allow it to boot up. Eventually you'll be greeted with the main pfSense menu on the console with 16 options. Select option 1, Assign Interfaces to run the interface assignment wizard. |||||||||||||||||||| |||||||||||||||||||| The WAN interface needs to be assigned to the Bridged network adapter (Network Adapter, the Bridged network). The LAN interface needs to be assigned to our Host-Only adapter (Network Adapter 2, the Management network).Finally, OPT1 needs to be assigned to our internal network adapter (Network Adapter 3, the IPS 1 network). In my case, em0 became the WAN interface,em1 the LAN interface, and em2 was mapped to the OPT1 interface. Be sure to pay attention to the MAC addresses for em0, em1, and em2, and correlate that to the MAC addresses you documented earlier for Network Adapter, Network Adapter 2, and Network Adapter 3. After confirming that the settings are correct, you should automatically be returned to the pfSense main menu. Now that we have assigned the network interfaces, we have to configure IP addresses for these interfaces. In most cases, the WAN interface (Bridged network) will automatically get an IP address from the device on your physical network that provides DHCP services. If this is not happening, you may have to |||||||||||||||||||| |||||||||||||||||||| troubleshoot your physical network. This is beyond the scope of our guide. As for LAN and OPT1, we have to manually set the IP address, subnet mask, and DHCP scopes for these networks. Select option 2 in the pfSense main menu to get started. The configuration wizard starts by asking for the interface the settings should be changed for. We have to go through the process two times, for the LAN and the OPT1 interface individually. Here are the settings I recommend: LAN IP: 172.16.1.1 LAN Subnet bit count: 24 LAN upstream gateway address: LAN IPv6 address: Do you want to enable the DHCP server on LAN? y LAN DHCP start address: 172.16.1.10 LAN DHCP end address: 172.16.1.254 Do you want to revert to HTTP as the webConfigurator protocol? N OPT1 IP: 172.16.2.1 OPT1 Subnet bit count: 24 OPT1 upstream gateway address: OPT1 IPv6 address: Do you want to enable the DHCP server on OPT1? y OPT1 DHCP start address: 172.16.2.10 OPT1 DHCP end address: 172.16.2.254 Do you want to revert to HTTP as the webConfigurator protocol? n The reason we choose 1.10 and 2.10 as the DHCP start addresses is to reserve a few IP addresses for static DHCP allocations. Static DHCP allocations allow you to configure a DHCP server to always serve the same IP address |||||||||||||||||||| |||||||||||||||||||| when requested from a particular MAC address. We'll talk about this a little bit more later. For now, we are done mucking around in the pfSense CLI. From here on out, we will be using the webConfigurator to manage and configure pfSense. Note: If you are using 172.16.1.0/24 or 172.16.2.0/24 in your physical network, choose another network range to assign to the LAN and OPT1 interfaces to avoid network conflicts. Note : When you are configuring the LAN and OPT1 interface IP addresses, the configuration script will ask Do you want to revert to HTTP as the webConfigurator protocol? Always say no to this. This allows the web UI to accept HTTP logins, and makes your firewall credentials vulnerable to sniffing over the network. Your pfSense main menu should look something like this when you're all done: If so, let's move on to setting up pfSense from the webConfigurator. Before doing so however, you must configure the Host-Only adapter (aka the Management network) on your host. If you are using Windows as your host, and If you haven't already, you'll want to visit the Unbinding Network Protocols on Windows Virtual Adapters at a minimum. This document will guide you through configuring theVMnet1 Adapter on Windows with an IP address as well as unbinding network protocols on this adapter to increase the security of your hypervisor host. You may also want to consider Using Windows Firewall to Limit Exposure of Windows Hypervisor Hosts to further enhance the security of the Windows host when interacting with your lab VMs. Linux users can configure the IP address of their host-only network card by using the command utilities ifconfig orip addr add. Try one of the following |||||||||||||||||||| |||||||||||||||||||| commands to set the IP address of thevmnet1 interface: ifconfig vmnet1 172.16.1.2 netmask 255.255.255.0 ip addr add 172.16.1.2/24 dev vmnet1 Please note that you will need root permissions to run these commands, and that these settings will NOT persist between reboots of your hypervisor host system. Configuring IP address persistence for your host OS is beyond the scope of this guide. 12.8 webConfigurator - Initial Setup On the host OS, open your favorite web browser (I prefer Firefox and navigate to (or the IP address you assigned to the LAN interface of your pfSense system). The default credentials for access to the web interface are admin/pfsense. If this is your first time logging in, pfSense takes you through a nice little setup wizard. I'll highlight some of the important things to take note of, and/or change as necessary: Set the primary and secondary DNS servers you plan on using. I typically use 8.8.8.8 (Google public DNS) and 4.2.2.2 (Level 3 public DNS). If you're using this lab at work, your workplace may have restrictions and you may have to use their DNS servers instead. Uncheck the checkbox next to Block private networks from entering via WAN rule. Uncheck the checkbox next to Block Bogon Networks on this page as well. The other default settings on the first time setup wizard should be fine. Please note that you will be asked to change the password for the admin account as a |||||||||||||||||||| |||||||||||||||||||| part of the setup wizard. Document the password, and keep it somewhere safe. Our next order of business is to restrict access to the web interface of the firewall to the machine with the IP address 172.16.1.2. On the menu bar at the top of the page, select Firewall > Rules. Click on LAN to modify the firewall policy for the LAN interface. Click the Add button with the arrow facing up. This will add the firewall rule to the top of the firewall policy to where it will evaluated first. The Edit Firewall Rule page is fairly straightforward. I've highlighted the options to be aware of. |||||||||||||||||||| |||||||||||||||||||| When you are done, click the Save button at the bottom of the page. This will take you back to the previous page. Click on the Apply Changes button to save your new firewall rule. Next, we want to an alias. Navigate to Firewall > Aliases. On the Firewall Aliases page, click on IP, then click Add. Create an alias with the following settings: |||||||||||||||||||| |||||||||||||||||||| Click Save to be brought back to the previous page, then click Apply Changes. We just created an alias for RFC1918 networks (local networks that are not routable through the public internet) This will come in handy later for creating firewall rules around these networks. Our final action for now is to navigate to System > Advanced, and click to fill in the checkbox next to the option Disable webConfigurator anti-lockout rule, and click Save on the bottom of the page. Be aware that if you did not create the firewall rule above, this will lock you out of the webConfigurator as soon as you click Save. 12.9 Take a Snapshot Snapshotting is a VERY important concept with VMs. It lets you restore your virtual machines to its state in the past when the snapshot was taken. This allows you to undo misconfigurations, solve problems or potential issues that might affect the VM, in relatively easily fashion. We're going to take a snapshot of the pfSense virtual machine in its current state. In the VMware Workstation main screen, in the Library pane, right click on pfSense, click Snapshot > Take Snapshot... |||||||||||||||||||| |||||||||||||||||||| A new window pops up, offering two input boxes: one to name the snapshot, and another to enter a description. I named the snapshot "baseline" and gave a brief description of what the snapshot is for. If you keep multiple snapshots make sure to use good descriptions that include WHEN the snapshot was taken, and WHAT STATE the VM is in when you took the snapshot. After you are finished, click the Take Snapshot button. VMware Workstation will begin generating the snapshot for your virtual machine immediately, and should be finished in moments. If you ever have to restore a VM from a snapshot, just right click on the virtual machine in the Library pane and navigate to the Snapshot sub-menu. You will be offered the option to either revert to the most recent snapshot, or to open the Snapshot Manager where you can choose a specific snapshot you want to revert to. The restoration will take a moment or two, and voila, your VM is |||||||||||||||||||| |||||||||||||||||||| back in state it was in when the snapshot was taken. 12.10 pfSense Summary At the end of all this, you should have a pfSense VM with the following settings: ● 512MB of RAM ● 5GB of disk ● 1 Virtual CPU ● DVD drive disabled ● Audio disabled ● USB controller disabled ● 3 Network interfaces: ○ 1 Interface connected to theVMnet0 (Bridged) network (WAN interface) ○ 1 Interface connected to theVMnet1 (Management) network (LAN interface) ○ 1 Interface connected toVMnet2 (IPS 1) network (OPT1 interface) pfSense should be configured as followed: ● The WAN interface should be the network adapter connected to the BridgedVMnet. This interface should automatically get an IP address via DHCP. If this isn't happening, get some help with whoever administers your local network, or start troubleshooting. Alternatively, configure a static IP address ● The LAN interface should have an IP address of 172.16.1.1. ● The LAN network should be 172.16.1.0 with a /24 (255.255.255.0) subnet mask ● DHCP should be configured for this network with a scope of 172.16.1.10-172.16.1.254 ○ 172.16.1.3-172.16.1.9 are available for setting static DHCP mappings ● The OPT1 interface should have an IP address of 172.16.2.1 ● The OPT1 network should be 172.16.2.0 with a /24 (255.255.255.0) subnet mask ● DHCP should be configured for this network with a scope of 172.16.2.10-172.16.2.254 ○ 172.16.2.2-172.16.2.9 are available for static DHCP mappings. ● Your host |||||||||||||||||||| |||||||||||||||||||| OS accesses pfSense from the Management interface. You should have one virtual adapter connected toVMnet1 (the one we will be using) and VMnet8 (which will NEVER be used. You can delete OR disable the interface in your host OS.). The VMnet1 adapter should have an IP address of 172.16.1.2. We created a firewall rule that serves as our anti-lockout rule for allowing the host OS to access pfSense via the web interface over HTTPS. ● You should have an alias for RFC1918 networks configured for use with the firewall policy. ● You should have at least one good snapshot you can revert to if you need to redo a step or somehow got lost/confused. 12.11 What's Next? Now that you have an operational pfSense VM, your next steps are to read the pfSense Firewall Rules and Network Services Guide. Make sure the firewall policies match those illustrated in the segmentation guide, and make sure that the pfSense VM is hosting all the services mentioned in the Core Network Services guide to make your firewall fully functional and ready to handle your lab network. 12.12 Final Connectivity Checks and Troubleshooting Once you have finished setting up your pfSense VM according to the instructions laid out, Open a console session to the pfSense VM, and attach your mouse and keyboard by clicking on the console. If the screen is blank, hit the Enter key to bring up the pfSense main menu. Select option 8) Shell to start a shell session on pfSense. The first, run the command ping -c 5 www.google.com |||||||||||||||||||| |||||||||||||||||||| The output from the ping command should look similar to the output above. Next, run the commandnslookup google.com Again, you should have output similar to the illustration above. Finally, runcurl -I The output of your curl command should be similar to the screen capture above. If so, you may typeexit to leave the shell, and close your connection to the pfSense console. If you had any problems with any of the commands above, you've got troubleshooting to do. When it comes to troubleshooting, start at layer 1 of the OSI model (Physical) and work your way up. Here are some questions to help the troubleshooting process: ● Which of the commands above failed? Did they all fail? |||||||||||||||||||| |||||||||||||||||||| ● Have you checked your physical cabling or wireless network connectivity? ● Does your company/LAN have MAC filtering/port security on? You might need to modify switch settings for your drop, or talk to your network administrator. ● Have you checked your upstream gateway/SOHO router? Trying pinging the gateway's IP address to see if your pfSense VM can reach the gateway. Try running the traceroute command to google.com to see if and where your ping command is failing. ● Do you have an upstream firewall in place? Is 443/TCP allowed outbound? Is 53/UDP allowed outbound? Is ICMP allowed? ● Did you input the commands correctly? Try entering them again. No, I'm not patronizing you. Once you have verified that your firewall's connectivity, policies, and services are properly configured, SNAPSHOT THE VM BEFORE moving on and trying your hand at setting up the remaining VMs. 12.13 Your Turn The pfSense virtual machine required a lot of custom configuration, both in VMware Workstation, as well as in the VM itself once the OS was installed. Now that we did that together, I'm going to have you create the rest of the virtual machines yourself -- with the exception of the Metasploitable 2 VM since it's a special case for reasons you will see momentarily. Below are a set of spec sheets to help you create the remaining virtual machines for the lab on your own. Think of them as checklists you should be able to run through unaccompanied. All of our Linux-based virtual machines are built on Debian Linux based distros (both, Ubuntu and Kali Linux are derived from Debian), so setup and configuration of all of our lab VMs should be very similar. 12.14 Kali Linux VM |||||||||||||||||||| |||||||||||||||||||| Kali Linux is a popular penetration testing distro. Popular because hackers and script kiddies are lazy, and practically everything you need for performing a penetration test or joining anonymous is installed by default. Kali is going to be our loud, obnoxious attacker that we're going to use as a noise generator for the express purpose of generating events on our IPS VM. I want you to perform the following tasks: ● Download Kali Linux 64-bit here: ● Create a VM with the following settings: ○ Configure the VM to install from the Kali Linux ISO you downloaded ○ Set the Guest OS family to Linux ○ Set the Guest OS version to Debian 8.x 64-bit ○ Name the VM "Kali" ○ Allocate 80 GB of space for the disk ■ Choose the Store virtual disk as a single file option ● Customize the following hardware settings ○ Allocate 4GB (4096MB) of RAM ○ Allocate 1-2 processors, where total number of cores is no more than 2 ○ Connect the VM's network adapter toVMnet2 ■ Ensure that Connect at power on is checked ○ Remove the USB Controller ○ Remove the Sound Card ○ Remove the Printer ● Install Kali Linux ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy ● After the install is completed, power off the VM and adjust the following settings: ○ Remove the CD/DVD drive ○ Make sure to document the MAC address under advanced settings for the network adapter. ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP, for the MAC address of adapter |||||||||||||||||||| |||||||||||||||||||| 1 for the Kali VM; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command, in this case, will work from this virtual machine, due to the unique firewall policy on the OPT1 network ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade ○ Create a snapshot for the VM 12.15 SIEM VM SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for log aggregator; we will be having our IPS VM log its events here. We're going to be running Splunk on this VM. Splunk is a commercial program for managing logs on a pretty large scale. By default, and with no licensing, Splunk only allows you to collect or index 500MB worth of logs per day however, there are /ways/ around this that we'll talk about later. I want you perform the following tasks to set up the SIEM VM: ● Download Ubuntu Server 16.04.1 LTS 64-bit here: |||||||||||||||||||| |||||||||||||||||||| ● Create a VM with the following settings: ○ Configure the VM to install from the Ubuntu Server ISO you downloaded ○ Ubuntu supports VMware's Easy Install option, but we're not going to use it ■ Fill out whatever data you want in the full name, username, password and confirm fields ○ Name the VM SIEM ○ Allocate 80 GB of space for the disk ■ Choose the Store virtual disk as a single file option ● Customize the following hardware settings ○ Allocate 4GB (4096MB) of RAM ○ Allocate 1-2 processors, where total number of cores is no more than 2 ○ Connect the VM's network adapter to Host-only (VMnet1) ■ Ensure that Connect at power on is checked ○ Remove the USB Controller ○ Remove the Sound Card ○ Remove the Printer ● Before powering on the VM to install Ubuntu Server... ○ Enter the Virtual Machine Settings under the Hardware tab. Remove the CD/DVD drive with the file namedautoinst.iso, as well as the Floppy drive with the file named autoinst.flp |||||||||||||||||||| |||||||||||||||||||| ● Install Ubuntu Server ○ Make sure to install Standard System Utilities and OpenSSH Server role during the Software selection phase ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy ● After the install is completed, power off the VM and adjust the following settings: ○ Remove the remaining CD/DVD drive |||||||||||||||||||| |||||||||||||||||||| ○ Make sure to document the MAC address under advanced settings for the Network Adapter ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP, for the MAC address of network adapter for the SIEM VM; assign it the IP address 172.16.1.3, making sure to enter a description that tells you which VM this static mapping belongs to ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade ○ Create a snapshot for the VM 12.16 IPS VM This VM is going to be responsible for running the AFPACKET bridge between the IPS 1 (VMnet2) and IPS 2 (VMnet3) virtual networks. Perform the following tasks to install the IPS VM: ● If you haven't already for the SIEM VM, Download Ubuntu Server 16.04.1 LTS 64-bit |||||||||||||||||||| |||||||||||||||||||| here: ● Create a VM with the following settings: ○ Configure the VM to install from the Ubuntu Server ISO you downloaded ○ Ubuntu supports VMware's Easy Install option, but we're not going to use it ■ Fill out whatever data you want in the full name, username, password and confirm fields. ○ Name the VM "IPS" ○ Allocate 80 GB of space for the disk ■ Choose the Store virtual disk as a single file option ● Customize the following hardware settings ○ Allocate 2GB (2048MB) of RAM ○ Allocate 1-2 processors, where total number of cores is no more than 2 ○ Connect the VM's first Network Adapter to Host-only (VMnet1) ■ Ensure that Connect at power on is checked ○ Create two additional network adapters. ■ Attach Network Adapter 2 toVMnet2 (IPS 1) ● Ensure that the Connect at power on checkbox is unchecked ■ Attach Network Adapter 3 toVMnet3 (IPS 2) ● Ensure that the Connect at power on checkbox is unchecked ○ Remove the USB Controller ○ Remove the Sound Card ○ Remove the Printer ● Before powering on the VM to install Ubuntu Server ○ Enter the Virtual Machine Settings under the Hardware tab. Remove the CD/DVD drive with the file namedautoinst.iso, as well as the Floppy drive with the file named autoinst.flp |||||||||||||||||||| |||||||||||||||||||| ● Install Ubuntu Server ○ Make sure to install Standard System Utilities and OpenSSH Server role during the Software selection phase ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy ● After the install is completed, power off the VM and adjust the following settings: ○ Remove the remaining CD/DVD drive ○ Check the Connect at power on checkbox for Network Adapter 2 and |||||||||||||||||||| |||||||||||||||||||| Network Adapter 3 (connected toVMnet2 and VMnet3) ○ Make sure to document the MAC address under advanced settings for all three network adapters, making sure to note which MAC address pairs up to which network adapter, connected to which VMnet ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP, for the MAC address of network adapter connected to VMnet1 (Management network, Host-onlyVMnet); assign it the IP address 172.16.1.4, making sure to enter a description that tells you which VM this static mapping belongs to ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server. ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense ○ Note: Theping command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will failUpdate/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade ○ Create a snapshot for the VM 12.17 Metasploitable 2 Metasploitable 2 is a little different from the other VMs. It is already precreated, we just need to import it, and reconfigure some of the virtual hardware. This should be pretty simple to do. First, download a copy of |||||||||||||||||||| |||||||||||||||||||| Metasploitable 2 from . Metasploitable 2 is distributed as a zip archive, so you will need a compression tool to extract it. On Linux/Unix systems, you can use the command line utilityunzip. On Windows, I recommend 7-Zip, which can be found at . Download the Metasploitable 2 archive, and decompress it in the download directory. You will be left with a folder labeledmetasploitable-linux-2.0.0. Inside of that folder is another folder labeled Metasploitable2-Linux. Move the entireMetasploitable2-Linux directory to the same directory you use to store your other virtual machines. In my case, this is D:\Workstation-VMs. In the VMware Workstation main window, on the menu bar, click File > Open... and navigate to the folder you just moved theMetasploitable2-Linux directory to. For me, that path was D:\Workstation-Vms\Metasploitable2Linux. Select theMetasploitable.vmx file in this directory. A new VM named Metasploitable2-Linux should appear in the Library pane of VMware Workstation. |||||||||||||||||||| |||||||||||||||||||| There are a few more settings we need to adjust before starting this virtual machine. Using what you know from setting up the previous VMs, perform the following actions in the settings menu, before attempting to boot it: ● Remove the USB Controller ● Remove Network Adapter 2 ● Remove the CD/DVD drive ● Reconfigure The Network Adapter from NAT toVMnet3 (IPS 2 Network) ○ Document the MAC address of this network interface ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP, for the MAC address of network adapter connected toVMnet3 (IPS 1 network); assign it the IP address 172.16.2.3, making sure to enter a description that tells you which VM this static mapping belongs to! Power on the Metasploitable 2 VM, make sure it is bootable. If so, take a snapshot of the VM, it's ready to go for now. You won't be able to verify whether the DHCP mapping works until AFTER the IPS VM is configured. |||||||||||||||||||| |||||||||||||||||||| 12.18 Next Steps The VMware Workstation initial setup is all but done at this point. However, you're not quite out of the woods just yet. Here is a checklist of tasks to complete: ● If you haven't completed the section Defense in Depth for Windows Hosts, and you are using Windows as a hypervisor host, I would very highly suggest doing so. ● While not strictly necessary, you may want to complete Remote Lab Management section, for the host OS of your choice (e.g. Windows Remote Access, Linux/Unix Remote Access), Enabling SSH on Kali Linux, and/or if you're lazy like me, the section on Securing root SSH access. This will allow you to remotely manage all of the Linux VMs much more easily than through the VMware Workstation console. ● You have to install IDS/IPS software on the IPS VM. You'll need to complete the IPS Installation Guide to learn how to do this with either Snort or Suricata as your IPS software of choice. ● The SIEM VM needs to have Splunk installed and configured. You'll need to complete the Splunk Installation Guide. ● You may want to consider reading the Automated Patching for Linux Lab VMs chapter, and implementing theupdater.sh script for your Linux VMs. ● Do you want some ideas on where to take your lab? Check out the chapter In Your Own Image, for some tips on how to mold your VM lab to better suit your needs. 13 Setup - VMware vSphere Hypervisor (ESXi) Note: This guide was made using VMware vSphere Hypervisor 6.0, Update 2 with free licensing, and assumes that users of this guide are this software version or higher. VMware vSphere Hypervisor (to which from now on I am going to refer to as ESXi, its traditional name) is a bare-metal hypervisor. As opposed to the hosted hypervisors covered in previous sections,. you need dedicated hardware to install ESXi on, and a second workstation or laptop to manage the |||||||||||||||||||| |||||||||||||||||||| ESXi server. Configuration, installation and setup are a little bit different too, compared to hosted hypervisors. Read on, pay attention, and you'll come out of this just fine. 13.1 Installation As mentioned in Hypervisor and Hardware Considerations, ESXi is pretty picky about the hardware it recognizes and installs on. And then, even if it supports your system's components, there may be cases in which not all features of the motherboard/chipset are fully supported. This isn't much of a problem if you are using commercial server solutions from a known hardware vendor (e.g. Dell or HP servers), but if you're trying to build your own system (also known as a white box server) you may face some problems. For instance, I built a server based on a Supermicro motherboard that was supported by ESXi, but the Intel RAID controller wasn't. The server would recognize hard drives, but not RAID arrays. Others have told me about experiences where their motherboard was supported, but the integrated ethernet cards were not, so they were required to buy PCI/PCI-e network cards in order to run ESXi. Some people have managed to install ESXi on some pretty exotic hardware Shuttle PCs, Intel NUCs, among other things. A lot of these builds require recreating the installation ISO to ensure that drivers for specific chipsets/hardware components are included in the image file. Unfortunately, this is beyond the scope of this guide. I can't guarantee your success if you try to install ESXi on exotic hardware, but if you want to build your own server (instead of buying a new or used commercial system), here are some things to consider: ● ESXi works really well with Intel chipsets and network cards. There's a good chance that if you have an Intel chipset on your motherboard and/or Intel network cards (integrated or PCI cards), that you will do just fine. ● While most prefer Intel chipsets and network cards where possible, the next best solution is to utilize Broadcom network cards. A lot of sysadmins I know dislike Broadcom due to their loose interpretations of various standards, but support for Broadcom NICs in ESXi is pretty decent. |||||||||||||||||||| |||||||||||||||||||| ● VMware has a hardware compatibility list you can search here: . Generally, the compatibility list is very cumbersome to use. A better solution would be to Google search for "part/model number" "esxi" to see if there are any results for compatibility problems for the server model or the server components you want to use. So now that we got hardware compatibility out of the way, you need to actually download ESXi and install it. In order to do so, you need to create an account on VMware's website. After you have signed up and logged in, navigate to Downloads and find VMware vSphere Hypervisor(ESXi). VMware changes links around all the time, but for the time being, the page for ESXi can be found here: . On the download page, you will be presented with a free product key for the free version of ESXi. While there are some significant differences between the free and the enterprise versions of ESXi, for our purposes, the free version and licensing will work just fine. Record your license key, since we will need it later. Make sure to download the latest version of ESXi (which as of this writing is 6.5), the latest version of VMware vSphere Client (if you are using Windows as your hypervisor manager, and don't want to use ESXi's web-based UI). vSphere client can be found at |||||||||||||||||||| |||||||||||||||||||| So now that you have an ISO, you need to figure out how you're going to install ESXi on your server. Traditionally, you would use a CD burning utility to get the ISO onto a CD or DVD, and boot from the disk. However, many PCs and servers are lacking optical drives these days The alternative is to install ESXi from a USB thumb drive or other external media. I found a great guide on how to create a USB installer for ESXi here: With a bootable CD/DVD or USB drive prepared, there are multiple different options where ESXi can be installed to. IT professionals have been found to install ESXi on USB drives or even SD cards. The reason for this is that ESXi itself only requires around 4 GB of space to install properly, with most of the components of a running ESXi residing in memory, so a flash drive or SD card more than serves the purpose of booting the system. Installing on external media saves space on the drives you'll be using to store virtual machines, and also ensures that running VMs won't be contending for disk I/O with ESXi itself. If you don't have a spare SD card or USB drive laying around, don't worry about this too much, just bear in mind that this is an installation option |||||||||||||||||||| |||||||||||||||||||| available, and that a lot of sysadmins do this. Also remember that if you do decide to install ESXi to an SD card or USB drive, system logging will be disabled by default! To enable logging for this installation/boot option anyway, you can adjust the advanced host settings of the ESXi to use one of the datastore disks available (where VMs are stored) to write its logs on, or specify the IP address of a syslog server where the logs should be sent to. The installation itself is pretty straightforward. During the process, you will be asked to set the root password for the system, create datastores on attached hard drives (for actually storing your virtual machines and their files), as well as configure the management interface (e.g. the network interface and IP address you will be using to connect to this server). If at all possible, you should consider setting a static IP address for this host, or creating a static DHCP mapping to ensure that your ESXi server always has the same IP address in the event the system reboots. Spending a portion of your day to figure out that DHCP has granted your ESXi server a new IP address is pretty annoying. 13.2 Accessing ESXi Once you have ESXi installed, and can confirm that it is reachable over the network (e.g. by usingping), you should be able to manage the system remotely. If you're using a Windows host, you have the option of utilizing the VMware vSphere Client to access your ESXi server. Simply plug in the IP address of your ESXi host, enter a username and password (in this case, you want to log in as the root user you created during installation), and click the Login button. |||||||||||||||||||| |||||||||||||||||||| If you are not using a Windows OS, or you would rather not install/use the vSphere Client, you If you are not using a Windows OS, or you would rather not install/use the vSphere Client, you based web interface. The web UI can be used on Windows, OS X, or Linux, provided you are using a modern web browser that is able to handle HTML5 content. For the sake of simplicity, we will be using the HTML5 web interface to interact with the ESXi server and perform the configuration tasks, except where noted to do otherwise and/or if workarounds are required. Please keep in mind that the HTML5 web interface is still relatively new, and prone to bugs here and there. |||||||||||||||||||| |||||||||||||||||||| When you have successfully logged in to the server, you should be greeting with a dashboard that, among other things, shows you the list of VMs available on your ESXi instance. In your case, you shouldn't have any installed yet. In my case, since I'm literally running a web server out of my basement, a virtual machine named BlindSeeker is in the inventory list for this ESXi server. |||||||||||||||||||| |||||||||||||||||||| Get used to this interface. Know it and love it, since we'll be spending a lot of time here. The window pane on the left, entitled Navigator, is going to be your best friend. As the name implies, it is used to navigate the various settings available for managing the ESXi server. From checking resource utilization, to creating VMs, to enabling/disabling features on the ESXi host, the Navigator pane is the key to getting around. 13.3 Hypervisor Setup ESXi is a little bit different from our hosted hypervisors. We need to perform some initial configuration and a couple of post-installation tasks in order to make it ready for us to get started. 13.4 Licensing On the Navigator pane, under Host, there is an option called Manage. When you click on it, a series of tabs is displayed on top of the middle pane, each |||||||||||||||||||| |||||||||||||||||||| related to different aspects of managing the ESXi host. Select the License tab, then click Assign License. Input the key we got from the VMware website earlier. By default, ESXi will start up in what is called evaluation mode, which offers you the full featureset available for the system (which equals a vSphere Enterprise Plus license) for 60 days. As you need to enter a valid license key at this point at the latest to preserve functionality, it is best to get that taken care of up front, rather then letting it become a problem later. If your key is valid and has been entered successfully, your Licensing page should look like this. Please note that I intentionally blanked out my license key on this illustration. 13.5 Networking and Virtual Switches ESXi's virtual networking is based off of creating virtual switches, then setting up port groups on these vSwitches, with a VM's virtual network adapter being assigned to a port group. There are no NAT networks, host-only networks, or internal networks here; just virtual switches and port groups. To provide access to a physical network, a virtual switch can be uplinked to one or multiple physical network port(s) on the ESXi host machine to serve the same purpose as a bridged network. Without an uplink, a stand-alone vSwitch operates very similarly to a private or host-only network. To do networking on our ESXi server the right way, we need at least two ports available on the server itself that are connected to the physical network. One of the network cards/ports is reserved by the ESXi host's VMkernel port (linked to a device named vmk0) for managing the ESXi server itself, while the other network card will be dedicated to network traffic to and from our VMs. To verify that your network cards/ports have been recognized by ESXi properly, under Navigator in the systems web UI, click Networking, then select the Physical NICs tab. ESXi will display the status of each recognized network card available to the system, including whether or not the cable is |||||||||||||||||||| |||||||||||||||||||| plugged in and/or what link speed the network card has negotiated to. If your network card(s) are NOT showing up here, ESXi does NOT have drivers for them or they are not supported. In my case, I have four network ports (vmnic0vmnic3), two of which are using the e1000e driver (Intel), and two of which using the bnx (broadcom) driver. Additionally you can see that two of the network ports are physically connected and have a link speed of 1gbps each. Don't have two network cards? Don't fret. There are ways to force the VMkernel port and the vNICs of the virtual machines to use the same network card/port. Though I must state that this goes against best practice, is not technically supported, and may not be an option in future versions of VMware ESXi. I'll show you how to do it anyhow, because this is a lab environment and you shouldn't be hosting anything of importance here anyway, right? Right. Now, let's create some virtual switches. 13.6 Creating Virtual Switches Under the Navigator pane, click on Networking, then click on the Virtual switches tab. You'll notice that a device named vSwitch0 is already present. This is the virtual switch that ESXi creates by default to attach the VMkernel port to the management interface, and we won't be using it. If you were to click Edit Settings for this vSwitch, you would see that this vSwitch is uplinked to the network card you chose as the management interface when you configured networking during the initial ESXi installation. For now, click the Add standard virtual switch option. |||||||||||||||||||| |||||||||||||||||||| Name this vSwitch "Management". Click the X on the right side of the Uplink 1 field to remove the drop-down. We do NOT want this virtual switch to have an uplink. Click the Add button to create this vSwitch. Next, click the Add standard virtual switch option again. This time, name the virtual switch "IPS 1", and as you did with the Management virtual switch, remove the Uplink 1 field by clicking the X next to the

drop-down. Click the triangle next to Security to expand the corresponding settings for the virtual switch. Click the Accept radio button for all three settings - Promiscuous mode, MAC address changes, and Forged transmits. To finish, click the Add button. |||||||||||||||||||| |||||||||||||||||||| Add another virtual switch with the exact same settings (all of the Security radio buttons set to Accept), and name it “IPS 2”. We need to create one more virtual switch. This one is going to serve as the Bridged network, hence we need to connect it to one of the host’s physical network cards/ports. Since one network port is already uplinked to vSwitch0, you need to ensure that you have at least one additional network connection available and physically connected. Note: If this isn’t feasible or just not an option for you, you may be able to use a the single NIC attached to vswitch0 as your Bridged Network vswitch, however it is highly recommend to have a secondary NIC attached to your physical network, and to create another vswitch associated to that network interface. Ahead of creating the vSwitch for our Bridged network, we need to confirm which network card vSwitch0 is uplinked to. To do so, select the Virtual switches tab, then click on vSwitch0.. On top of the overview page for this virtual switch, click on Edit settings. Take notice of the network interface listed in the Uplink 1 field. Click cancel to exit. |||||||||||||||||||| |||||||||||||||||||| Remember how I had you review which of your server’s physic al NICs actually were connected under the Physical NICs tab? Your listing of physical NICs might look somewhat different, but in my case,vmnic0 and vmnic3 were connected to my physical network. Thanks to the illustration above, we know thatvmnic0 is being used as uplink on vSwitch0. Navigate back to Networking > Virtual switches and click Add standard virtual switch. Name this virtual switch “Bridge”. This time, we’re going to keep Uplink 1, setting it to a vmnic interface that is physically connected to your physical network, but different from the one used on vSwitch0. In my case, I am using vmnic3. When finished, click the Add button. Back in the Virtual switches tab, you should have a list of vSwitches that looks something like this: |||||||||||||||||||| |||||||||||||||||||| 13.7 Port Groups Port groups are required to allow a VM’s network interface(s) to be attached to a virtual switch. Port groups also allow you to logically separate groups of virtual machines connected to the same virtual switch using VLANs. For example, you could have a port group assigned to an “Engineering VLAN”, and another port group assigned to a “Human Resources VLAN”. They would reside on the same ESXi server, they would use the same vSwitch, but they would be on different VLANs and are thus segmented from each other. There's more to segmentation than that, but that is outside the scope of this document. Depending on whether or not ESXi’s web interface is giving you problems, (See Resolving Some Interface Bugs) you may need a Windows system with the VMware vSphere client installed to configure port groups and virtual networking.. There was a bug where the Port groups tab under Networking will not display properly during initial setup, not without at least one virtual machine port group defined. If you did not download vSphere Client from the VMware website, you can get it via the ESXi server by opening up your web browser and pointing it to the management IP address of the host. For example, if your ESXi server’s IP address is 192.168.1.20, enter into your web browser, then click on the Download vSphere Client for Windows link to download the software from VMware’s site. 13.8 Adding Port Groups via the ESX Web Interface Under the Navigator pane on the web interface, click on Networking, select the Port groups tab, and click Add port group. On the window that pops up, name the port group Bridged and in the Virtual switch drop-down, select Bridge.When you are done, click the Add button. |||||||||||||||||||| |||||||||||||||||||| You should be back at the Port groups tab. Repeat this process three more times, creating a port group for the Management, IPS 1 and IPS 2 vSwitches. Your Port groups list should look like this when completed: 13.9 Resolving Some Web Interface Bugs Note: As of ESXi 6.0 Update 3, it appears some of the web interface issues I described below has been resolved. But to the benefit of readers everywhere, I’m including this section here on work-arounds to potential web UI issues, in case you run into new, or similar bugs of your own. If you’re running ESXi 6.0 update 3 or above, you should be able to skip this section. While making this guide, I ran into some web interface bugs related to configuring virtual switches and port groups for some reason, I did not have the option to create port groups This issue appears to affect ESXi 6.0 up to and including ESXi 6.0 Update 2. Fortunately, there are workarounds available, especially if your management workstation runs the Windows OS and you have the vSphere Client installed. For the sake of keeping your options open, I’ll document workarounds to web interface issues I had to use initially, but I’m also going to show you how to update the ESXi embedded web interface using VMware Labs’ Flings to fix the problem. Special thanks to @JohnPittman for pointing me towards this solution. 13.10 VMware Flings Flings are experimental applications and tools made available for system |||||||||||||||||||| |||||||||||||||||||| administrators and virtualization enthusiasts to try out before they get accepted as official offerings or mainline additions to VMware products. If you try to download anything on Flings, they make you click a checkbox that says you agree to not install these things on production systems or hold VMware accountable of something terrible happens. In a nutshell, Flings are the equivalent of doing public beta testing of new features and applications. I’m going to walk you through installing the latest ESXi Embedded Host Client Fling. First and foremost, you need to enable access to the ESXi shell. By default, VMware disables access to the command line shell because more often than not, the average user doesn’t really have a compelling reason to be mucking around in there, unless they’re working with VMware support, or doing strange things to the OS (kinda like what we’re about to do). So from a security standpoint, it makes sense. No need to worry however, enabling this access is trivial. In the web interface, click on Host under Navigator. In the middle pane, click on Actions. On the drop-down menu, select Services, then click on Enable console shell. This enables access to the shell interface. If you planned on connecting a mouse and keyboard to the machine running the ESXi software, or were going to use a lights-out management system (LOM) or IPMI viewer application of some sort, this is the only service you need to enable. Apart from that, you may have noticed the menu item Enable Secure Shell (SSH) under Services. In addition to enabling the console, repeat the steps laid out above once more, only this time this option to start the SSH service on your ESXi host on port 22. From here on, you can use your favorite ssh client, along with the root username and password, to connect to your ESXi server over SSH. If your SSH login is successful, and console access is enabled, you should be greeted with this prompt: |||||||||||||||||||| |||||||||||||||||||| Otherwise, if you prefer using keyboard and mouse, LOM, or IPMI, hit alt+f1 in the ESXi console main menu to gain access to the shell login prompt, and hit alt+f2 to return to the main screen. If console access is enabled, you’ll be greeted with a login prompt: With CLI access enabled on the ESXi host, go to and verify the current version and name of the file we will be installing. In my case, the filename isesxui-signed-4393350.vib, its version being 1.9.1. Now that you have confirmed that you are going to use the latest version of the esxui package, be sure to copy the download link listed on the page. For instance, . Copy it to your clipboard or to the text editor of your choice, then return to your ESXi console or SSH session. In the CLI, we need to run the following command: esxcli software vib install -v [url here] The complete command I entered looked like this: esxcli software vib update -v If the command completes successfully, you get a pair of single quotes (’’) and the prompt returns to you. If you get any errors, following some basic troubleshooting procedures might prove helpful, e.g. making sure that the ESXi server has a DNS server and a default gateway configured, it is allowed to use HTTP outbound, in case of a proxy installed on the network that the ESXi host is configured to use it, etc. |||||||||||||||||||| |||||||||||||||||||| If the system cannot connect to the internet for known or unknown reasons, you need to use the offline installation package and/or employ some method of copying the file to the server. Assuming that you enabled SSH, using SCP would be the easiest way to get the file onto the server. If you need further guidance on installing the esxui package, go here: . If you get an error message stating that VIBs have been skipped, this means that the package has already been installed. You can confirm that the package has been successfully installed by running the following command: esxcli software vib list | less Look for esx-ui in the Name column, and verify that the version number displayed matches the one of the file you previously downloaded from the VMware labs site. Once you have verified the package has been installed, exit your console session. You have to logout and log back in to your web UI session as well. After logging back in, navigate back to the Host > Actions > Services and disable the console and/or SSH services as necessary. The console and/or SSH services should not remain enabled. |||||||||||||||||||| |||||||||||||||||||| 13.11 What if I don’t want to use experimental software? Some of you might need to make your lab environment production quality, and unfortunately, most of the time, that means that you can’t install untested/unvetted software upgrades at will, so VMware Flings might be out of the question for you. The only other officially supported fallbacks for administering ESXi servers are using VMware’s vCenter Server product (which requires licensing and costs money), or to use the Windows vSphere Client. We will be focusing on using the Windows vSphere Client in our guide as a fallback. This guide will teach you how to use Windows system and the vSphere Client to create virtual machine port groups, then return to the ESXi web interface to create the remaining port groups we need for our lab. Open the Windows vSphere Client, input the IP address of your ESXi server, username (root) and password, and click Login. Note that the client will likely give you an SSL cert warning if you are using ESXi’s defaults. Accept the warning and allow the connection. |||||||||||||||||||| |||||||||||||||||||| Once you have successfully logged in, the vSphere Client should display an interface with a series of tabs in the center pane for controlling the ESXi host. Select the Configuration tab. |||||||||||||||||||| |||||||||||||||||||| On the left side of the Configuration tab, you find a pane labeled Hardware. In this pane, click on Networking. A list of all the virtual switches and their logical network configuration appears. To start creating our initial port group,click on Properties… to the right of Standard Switch: Bridge. If you don’t have a Bridge vSwitch as there is no second physical network card/port available on your server, use vSwitch0 to create your initial port group instead. |||||||||||||||||||| |||||||||||||||||||| A new window appears, showing you the properties for the Bridge vSwitch (or for vSwitch0, respectively). Click on the Add… button on the bottom left corner to start the Add Network wizard. |||||||||||||||||||| |||||||||||||||||||| On the first screen, under Connection Type, select the radio button in front of Virtual Machine, then click Next. On the Connection Settings screen, in the Port Group Properties section, change the Network Label: input box to Bridged, then click Next again. The final page will confirm your settings. Click Finish. |||||||||||||||||||| |||||||||||||||||||| Close the Bridge Properties window (or the vSwitch0 Properties window, if that is where you just created the port group on). Back under Configuration > Hardware > Networking, the new port group should appear on the Bridge virtual switch (or on vSwitch0, respectively). If you’d like, you could repeat the process and create the remaining port groups we need on the other virtual switches using the vSphere Client as well, but with the first virtual machine port group generated, this can be done via the ESXi web interface, too. 13.12 Final Flight Check At this point, you should have ESXi installed. You should be able to access the web interface from a management workstation running any operating system, so long as you have a modern web browser. You may or may not be using a Windows system as your management workstation for managing and configuring the ESXi host, but if you are, you installed the vSphere Client is installed in case you run into web UI issues.. You should have have 5 virtual switches (four, if you only have a single physical network port available), and at least 5 port groups (including the default Management Network port group used for managing ESXi). |||||||||||||||||||| |||||||||||||||||||| You’ll want to make sure that ESXi recognizes all the RAM installed in your server, and that all of the disks (or RAID arrays, if you’re fancy) attached to your ESXi server are available as well. Quickest way to do this is by checking the host’s Hardware status section. To get there, fire up the ESXi web interface, log in, and under the Navigator pane, select Host. The status report titled Hardware can be found in the center pane: The Hardware overview provides information on how much RAM, how many network cards/ports, and how many CPU cores were detected by the system. Additionally, it displays the network settings of the actual ESXi host (IP address, gateway, DNS, etc.). Finally, it lists all the configured port groups, and all of the connected data stores (disk drives we can use for storing VMs and other files). |||||||||||||||||||| |||||||||||||||||||| In the upper right corner of the center pane, there are a set of bars that give a quick at a glance view of the resource utilization on the server. For instance, you can see that I have approximately 1.9TB of space available, with a little over 100GB of space used, I have 32GB of RAM in total, with 1.86GB in use, and CPU usage is minimal at 111mhz out of an available 12.4GHz (3.10 Ghz x 4 cores). This system is as ready as it is going to get for hosting VMs. If for some reason you think that not all of your server’s storage space is detected or utilized, consider visiting the Storage interface under the Navigator pane. Try to detect if for some reason your array or disks simply weren’t formatted or converted to datastores by click the Devices tab. If there are some uninitialized disks or disks with free space listed, then format and initialize the disks as needed. In regards to other hardware-related issues, make sure all your hardware is seated and installed properly. In case your network cards haven’t been recognized, you may have to acquire new ones that are supported by ESXi’s driver base. If RAM or CPU is not recognized by ESXi, verify that the hardware is detected by the system BIOS; it might turn out to be as simple as a hardware malfunction, or enabling a setting in the BIOS. If everything in your server shows up properly and you have all the resources you need, let's move on. 13.13 Creating the First VM, pfSense |||||||||||||||||||| |||||||||||||||||||| pfSense is the keystone holding this entire configuration together. Personally, it's my favorite firewall distro due to ease of use, the amount of functionality it includes out of the box, combined with a plugin/add-on system for additional functionality. If you have the CPU, RAM, and disk, pfSense can easily be converted into a socalled “Next-Generation” firewall. To start, make your way to . Download the latest installation ISO for the amd64 architecture .Once you have downloaded the ISO, depending on what operating system you are using to manage your ESXi server, you may need to acquire a compression utility, as its maintainers distribute pfSense as an ISO image file compressed with gzip. This means that we’ll need to decompress the ISO file, using for example gunzip (Mac/Linux) or 7-Zip (Windows). To make it available for use, we have to upload the decompressed installation image file to one of the datastores of our ESXi server. On the web interface, under the Navigator pane, click on Storage, and ensure the Datastores tab is selected. In the displayed list, right click on the datastore you wish to store the ISO on, then click Browse. In my case, I chose the datastore named “Rogue 2”. In the Datastore browser window that appeared, click on the Create directory option. As we’re going to store all of our Linux and BSD ISOs in this directory, let’s name it “Linux_and_BSD_ISOs”. Click the Create directory button. |||||||||||||||||||| |||||||||||||||||||| Back in the Datastore browser, select the newly created directory, then click Upload in the upper left portion of the window. This will open a file browser on your local machine. Navigate to where you downloaded the pfSense ISO. If you haven’t already, make sure that the ISO has been decompressed. After selecting the ISO file, the process of uploading it to the ESXi server’s datastore is started. On completion, you should be able to select the ISO file in the Datastore browser window to display its file statistics. |||||||||||||||||||| |||||||||||||||||||| After successfully uploading the ISO file to the desired datastore, close the Datastore browser window. 13.14 Adding a New VM Now that we have the ISO ima ge to boot from ready, let’s create the VM for pfSense. On the ESXi web interface, click on Virtual Machines under the Navigator pane. This brings you to a list of currently registered VMs on the server. On your system, this list should be empty ; in my case, a virtual machine named BlindSeeker (my previously mentioned web server) is already available. Click on Create/Register VM to start the New virtual machine wizard. On the first screen, select the option Create a new virtual machine, then click the Next button. |||||||||||||||||||| |||||||||||||||||||| The next page asks to provide a name for the VM, and choose the family and the version of the OS that will be used later. Enter “pfSense” in the Name field, choose Other from the OS family drop-down, select FreeBSD (64-bit) as the OS version , then click Next. |||||||||||||||||||| |||||||||||||||||||| The next section offers a list of datastores available on the ESXi host. From that list, select the one where the files related to the virtual machine should be stored. Click on the datastore of your choice to select it (if you have more than one datastore), then click Next to continue. |||||||||||||||||||| |||||||||||||||||||| The next page allows us to make various customizations to our virtual machine’s hardware. The first and most obvious settings displayed are CPU, Memory and Hard disk space. For the pfSense VM, adjust the the corresponding values to 1 CPU core, 512 MB of RAM, and 5GB of disk space. As we change the configuration how the disk space is provided to the virtual machine, click the triangle next to Hard disk 1 to bring up the additional hard disk settings. In the Disk Provisioning section, select the radio button next to Thick provisioned, lazily zeroed, and in the Virtual Device Node section, choose SATA controller 0 from the drop-down on the left. The one to the right should read SATA (0:0). |||||||||||||||||||| |||||||||||||||||||| Next, in the SCSI Controller 0 field, click the X on the right to remove this device entirely as it isn’t needed. As we need additional NICs on this VM, click twice on the button at the top of the window named Add network adapter. Two more network cards appear right below Network Adapter 1, each of them labeled New Network Adapter. Configure the first network adapter to connect to the Bridged port group, the second one to connect to the Management port group, and the third NIC to connect to the IPS 1 port group. Ensure that the checkbox labeled Connect is checked for all three adapters. |||||||||||||||||||| |||||||||||||||||||| Finally, click the Add other device option at the top of the window, and from the appearing dropdown menu, select CD/DVD drive. An entry labeled New CD/DVD Drive should appear in the overview, immediately underneath the network adapters. Click on the triangle next to this text to bring up the settings for the virtual CD/DVD drive. Ensure that the checkbox Connect at power on is checked, then in the dropdown field that is most probably set to Host device, change this value to Datastore ISO file. When doing so, the Datastore browser window should open up. Navigate to the location you uploaded the pfSense ISO to, and select the file. Afterwards, click the Next button to move on to the final screen of the wizard. This is a confirmation page, allowing to review all the settings selected for this VM. Your settings should look similar to this: |||||||||||||||||||| |||||||||||||||||||| Click Finish and allow it some time for ESXi to create the new virtual machine. On completion, you should be returned to the virtual machines listing, and the new VM should be displayed there. Note: I realized that, for whatever reason, ESXi appears to reverse the order of the network adapters, at least in regard to what one would expect with the |||||||||||||||||||| |||||||||||||||||||| hardware overview from the New virtual machine wizard in mind. If you right click on the VM you just created, and select Edit Settings, you may find that Network Adapter 1 is set to IPS 1 as its port group, Network Adapter 2 is set to Management and Network Adapter 3 is set to Bridged. In fact, it appears as though ESXi considers the most recently created network adapter to be Network Adapter 1, then 2, and so on. For the setup of the virtual machine, this ordering doesn’t really matter. To fix this, simply swap the port group settings for Adapter 1 and Adapter 3, then click the Save button. The only other VM with multiple network adapters that we are going to create, is the IPS virtual machine. So don’t worry too much about it, just keep this in mind. 13.15 Installing pfSense pfSense is really easy to install. Start the virtual machine by navigating to Virtual Machines in the Navigator pane, then either right click pfSense and select Power > Power on from the menu, or click to highlight pfSense, and click Power on on the option bar above the VM list. In either case, the virtual machine should start up. Open a console window to interact with the VM. To do so, right click on pfSense in the virtual machines list again. This time, select Console > Browser Console to open a pop-up window that will act as a console to interact with the system running in the virtual machine. Click on this console lets you interact with the VM with your keyboard and mouse. To release the keyboard and mouse, use the key combination Ctrl + Alt. |||||||||||||||||||| |||||||||||||||||||| After booting the VM and attaching to the virtual machine’s console, select option 1, Boot Multi User [Enter] by hitting the enter key. Let pfSense boot up, and the system should automatically run the installer. Adjust your video, screenmap, and keymap settings as necessary, then select< Accept these Settings >. |||||||||||||||||||| |||||||||||||||||||| On the next screen, select < Quick/Easy Install > and let pfSense do all the heavy lifting. The next screen will inform you that the install will erase the contents of the hard disk. Since our virtual disk is already empty, this doesn’t matter in the least. Select OK, and let the installer run. |||||||||||||||||||| |||||||||||||||||||| The installer will go through and install pfSense to the virtual hard disk. At a certain point you will be asked whether the standard kernel or the embedded one should be used. Make sure to select < Standard Kernel >. Finally, the installer informs you to reboot the machine to boot from the hard drive. Allow the virtual machine to reboot in order to finish the installation procedure. After system has rebooted, in the ESXi web interface, close the console, right click on pfSense, and select Power > Power off to turn off the VM. 13.16 Final VM Settings In the Virtual Machines pane under Navigator, right click on pfSense and select Edit settings from the menu. In the section labeled CD/DVD drive, click the X to the right to remove the CD/DVD drive; we won’t need it anymore. |||||||||||||||||||| |||||||||||||||||||| Follow the instructions as described above for the CD/DVD drive to remove USB controller 1 as well: Afterwards, click the triangle next to Network Adapter 1, Network Adapter 2, and Network Adapter 3 to show the additional information available for these devices. Be sure to document the contents of the MAC address field, and which MAC address is connected to which port group. We will need this information in the next section to perform the network configuration portion of the pfSense installation. When finished, click the Save button to close the window. |||||||||||||||||||| |||||||||||||||||||| 13.17 Network Configuration Power on the pfSense VM, and allow it to boot up. Eventually you’ll be greeted with the main pfSense menu on the console with 16 options. Select option 1, Assign Interfaces to run the interface assignment wizard. To manage its network services and firewall policies, pfSense uses internal network interface descriptors (WAN, LAN, OPT1, etc.). Running the wizard and assigning the interfaces defines which of them (em0, em1, em2, etc.) corresponds to which of pfSense's internal descriptors. |||||||||||||||||||| |||||||||||||||||||| In the Final VM Settings section, we documented the MAC address of each network adapter and the respective ESXi port group it is connected to. With this information at hand, it should be easy to identify each of them on the list of valid interfaces the wizard displays on startup. After figuring out which pfSense interface corresponds to which ESXi network adapter(and port group), we need to assign em0, em1 and em2 to the WAN, LAN, and OPT1 descriptor accordingly. The WAN interface needs to be assigned to the network adapter connected to the Bridged port group, The LAN interface needs to be assigned to the network adapter connected to the Management port group, and finally, OPT1 needs to be assigned to the network adapter connected to the IPS 1 port group. In my case,em0 became the WAN interface, em1 became the LAN interface, and em2 became the OPT1 interface. After confirming that the settings are correct, you should automatically be returned to the pfSense main menu. Now that we have assigned the network interfaces, we have to configure their |||||||||||||||||||| |||||||||||||||||||| IP addresses. I very highly recommend configuring a static IP address for the pfSense WAN interface (using the Set interface(s) IP address menu option and configuring an IP address, subnet mask and gateway appropriate to your physical network) or creating a static DHCP mapping for the pfSense WAN interface on the device providing DHCP services to your physical network. We will be using the pfSense VM’s WAN IP address to route traffic from our management workstation (or jump box) to the lab network. In order to do this, the WAN interface’s IP address cannot change. for the LAN and OPT1 networks, we will have to manually set the IP address, subnet mask, and DHCP scopes for the networks. Select option 2 in the pfSense main menu to get started. The configuration wizard starts by asking for the interface the settings should be changed for. We have to go through the process two times, for the LAN and the OPT1 interface individually. Here are the settings I recommend: LAN IP: 172.16.1.1 LAN Subnet bit count: 24 LAN upstream gateway address: LAN IPv6 address: Do you want to enable the DHCP server on LAN? y LAN DHCP start address: 172.16.1.10 LAN DHCP end address: 172.16.1.254 Do you want to revert to HTTP as the webConfigurator protocol? N OPT1 IP: 172.16.2.1 OPT1 Subnet bit count: 24 OPT1 upstream gateway address: OPT1 IPv6 address: |||||||||||||||||||| |||||||||||||||||||| Do you want to enable the DHCP server on OPT1? y OPT1 DHCP start address: 172.16.2.10 OPT1 DHCP end address: 172.16.2.254 Do you want to revert to HTTP as the webConfigurator protocol? n The reason we choose 1.10 and 2.10 as the DHCP start addresses is to reserve a few IP addresses for static DHCP allocations. Static DHCP allocations allow you to configure a DHCP server to always serve the same IP address when requested from a particular MAC address. We’ll talk about this a little bit more later. Note: If you are using 172.16.1.0/24 or 172.16.2.0/24 in your physical network, choose another network range to assign to the LAN and OPT1 interfaces to avoid network conflicts. Note : When you are configuring the LAN and OPT1 interface IP addresses, the configuration script will ask Do you want to revert to HTTP as the webConfigurator protocol? Always say no to this. This allows the web UI to accept HTTP logins, and makes your firewall credentials vulnerable to sniffing over the network. Before we’re done using the command line, there is one more task we need to perform t o access the pfSense web interface. Select option 8 from the pfSense menu to open up a shell. Once you get a shell prompt, run the following commands: pfctl -d easyrule pass wan tcp 192.168.1.17 192.168.1.22 443 exit The first command, pfctl -d, temporarily disables the firewall on pfSense. This is necessary because by default, the firewall drops ALL connections from RFC1918 addresses on the WAN interface. More than likely, your pfSense vm’s |||||||||||||||||||| |||||||||||||||||||| WAN interface, and management workstation are going to be connected to a private network using RFC1918 addresses. For example, my home network uses the 192.168.1.0/24 private address range. pfSense assumes that the WAN interface will be exposed to the internet, so by default all RFC1918 addresses attempting to connect via the WAN interface are dropped (RFC1918 address ranges are NOT routable on the public internet). The management workstation you use will have to communicate with the pfSense VM through its WAN interface. Both of these network devices are on an RFC1918 network. Can you see where the problem is now? The first command we entered, pfctl -d, disables the firewall entirely. This is a temporary solution to allow us to connect to the firewall, through explicitly disabling the setting that blocks ALL RFC1918 addresses on the WAN interface. The second command, easyrule pass wan tcp 192.168.1.17 192.168.1.22 443, adds a firewall rule to the WAN interface of pfSense that allows 192.168.1.17 (the IP address of my management workstation) to access 192.168.1.22 (the IP address of WAN interface of the pfSense VM) on port 443 (https). Adjust the IP addresses of the management workstation, and the pfSense WAN interface in this rule as required for your network at work or at home. After you are done, type exit to leave the shell. Note: You will need to run both of the commands above before moving on to the next section. 13.18 webConfigurator - Initial Setup On your management workstation, open a web browser and navigate to the WAN address of the pfSense firewall (in my case, this is). The default credentials for access to the web interface are admin/pfsense. If this is your first time logging in, pfSense takes you through a nice little setup wizard. I’ll highlight some of the important things to take note of, and/or change as necessary: Set the primary and secondary DNS servers you plan on using. I typically use 8.8.8.8 (Google public DNS) and 4.2.2.2 (Level 3 public DNS). |||||||||||||||||||| |||||||||||||||||||| Be sure to uncheck the Block private networks from entering via WAN rule. This setting must be unchecked in order for your management system to be able to access the pfSense web interface (For a more detailed explanation, see the paragraphs at the end of the previous section, Network Configuration). You may also elect to uncheck the Block bogon networks checkbox (I typically uncheck this option since pfSense blocks traffic that isn’t explicitly allowed with a firewall rule). The other default settings on the first time setup wizard should be fine. Please note that you will be asked to change the password for the admin account as a part of the setup wizard. Document the password and keep it somewhere safe. Use a password manager if necessary. Next, on the menu bar at the top of the page, navigate to Firewall > Aliases. On the Firewall Aliases page, click on IP, then click Add. Create an alias with the following settings: |||||||||||||||||||| |||||||||||||||||||| Click Save to be brought back to the previous page, then click Apply Changes. We just created an alias for RFC1918 networks (local networks that are not routable through the public internet) This will come in handy later for creating firewall rules around these networks. Next, navigate to Firewall > Rules, and click on WAN, if it isn’t already selected. Your firewall rules for the WAN interface should look like this: That lone firewall rule? Never remove it. It is the one created by using the easyrule command, as described at the end of the previous section, Network Configuration. This rule is absolutely mandatory. Make sure to never create and place any other firewall rule for the WAN interface above this one. This rule is the only thing keeping you from being locked out of the firewall web interface. Once you are more familiar with the firewall rules UI, you may choose to edit this rule and change the description under the Extra Options section of this rule. Next, navigate to System > Advanced. Click to fill in the checkbox next to the option Disable webConfigurator anti-lockout rule, and click Save on the bottom of the page. This rule is meant to allow any machine connected to the LAN network to access the firewall’s web interface. Since we are going to manage pfSense by connecting to the WAN interface, and we just made sure that the necessary firewall rule is in place, disabling the antilockout rule is safe and reduces unnecessary exposure. One final task before moving on to the next section, we are going to make sure that the firewall is re-enabled. If no longer available, open a new console session to the pfSense VM in the ESXi web UI, and select option 8 from the firewall’s main menu to open a shell session. Enter the following command to re-enable the firewall: |||||||||||||||||||| |||||||||||||||||||| pfctl -e As a part of the initial webConfigurator setup, pfSense reloads the firewall config. While this SHOULD re-enable the firewall, running the command above ensures this has happened and that the firewall is operating normally. In case the firewall still was disabled, the console output will look like the one in the picture below:: This is the console output that is returned if the firewall has already been re-enabled: Make sure to verify that the firewall is re-enabled, and that you are still able to access the webConfigurator before moving on to the next section. 13.19 Take a Snapshot Snapshotting is a VERY important concept with VMs. It lets you restore your virtual machines to its state in the past when the snapshot was taken. This allows you to undo misconfigurations, solve problems or potential issues that might affect the VM, in relatively easily fashion. We’re going to take a snapshot of the pfSense virtual machine in its current state. On the ESXi web interface, click on Virtual Machines under the Navigator pane. In the Virtual machine list, right click on pfSense, then select Snapshots > Take Snapshot from the menu. A new window pops up, offering two input boxes: one to name the snapshot, |||||||||||||||||||| |||||||||||||||||||| and another to enter a description. I named the snapshot “Baseline” and gave a brief description of what the snapshot is for. If you keep multiple snapshots, make sure to use good descriptions that include WHEN the snapshot was taken, and WHAT STATE the VM is in when you took the snapshot. After you are done, click the Take Snapshot button. The ESXi server will begin generating the snapshot for your virtual machine immediately. You can monitor the progress in the Recent tasks pane at the bottom of the ESXi web UI window. To access previous snapshots, right click on the VM, and choose Snapshots > Manage snapshots. This opens a new window, displaying a tree view of snapshots available, and offers options for restoring from previous snapshots, deleting snapshots, etc. 13.20 pfSense Summary At the end of all this, you should have a pfSense VM with the following settings: ● 512MB of RAM |||||||||||||||||||| |||||||||||||||||||| ● 5GB of disk ● 1 Virtual CPU ● 3 Network interfaces: ○ 1 Interface connected to the Bridged port group (WAN interface) ○ 1 Interface connected to the Management port group (LAN interface) ○ 1 Interface connected to the IPS1 port group (OPT1 interface) pfSense should be configured as followed: ● The WAN interface should be the ESXi network adapter connected to the Bridged port group, meaning it should have an IP address on your physical network. Configure a static IP address or a static DHCP mapping for this interface to ensure that the address does not change. This is extremely important. ● The LAN interface should have an IP address of 172.16.1.1 ● The LAN network should be 172.16.1.0 with a /24 (255.255.255.0) subnet mask ● DHCP should be configured for this network with a scope of 172.16.1.10-172.16.1.254 ○ 172.16.1.2-172.16.1.9 are available for setting static DHCP mappings ● The OPT1 interface should have an IP address of 172.16.2.1 ● The OPT1 network should be 172.16.2.0 with a /24 (255.255.255.0) subnet mask ● DHCP should be configured for this network with a scope of 172.16.2.10-172.16.2.254 ○ 172.16.2.2-172.16.2.9 are available for static DHCP mappings ● Your management workstation should ble capable of accessing the pfSense webConfigurator on the WAN interface over HTTPS. We created a firewall rule that serves as anti-lockout rule for the webConfigurator on this interface. ○ Be aware that if the IP address of the management workstation or of the WAN interface changes, this firewall rule will need to be adjusted via the ESXi web UI, or in the console for the pfSense VM, using the easyrule utility ● You should have an alias for RFC1918 networks configured for use with the firewall policy. ● You should have at least one good snapshot you can revert to if you need to |||||||||||||||||||| |||||||||||||||||||| redo a step or somehow got lost/confused. 13.21 What’s Next? Now that you have an operational pfSense VM, your next steps are to read the pfSense Firewall Rules and Network Services Guide. Make sure the firewall policies match those illustrated in the segmentation guide, and make sure that the pfSense VM is hosting all the services mentioned in the Core Network Services guide to make your firewall fully functional and ready to handle your lab network. 13.22 Final Connectivity Checks and Troubleshooting Once you have finished setting up your pfSense VM according to the instructions laid out, Open a console session to the pfSense VM, and attach your mouse and keyboard by clicking on the console. If the screen is blank, hit the Enter key to bring up the pfSense main menu. Select option 8) Shell to start a shell session on pfSense. The first, run the command ping -c 5 www.google.com The output from the ping command should look similar to the output above. |||||||||||||||||||| |||||||||||||||||||| Next, run the commandnslookup google.com Again, you should have output similar to the illustration above. Finally, runcurl -I The output of your curl command should be similar to the screen capture above. If so, you may typeexit to leave the shell, and close your connection to the pfSense console. If you had any problems with any of the commands above, you’ve got troubleshooting to do. When it comes to troubleshooting, start at layer 1 of the OSI model (Physical) and work your way up. Here are some questions to help the troubleshooting process: ● Which of the commands above failed? Did they all fail? ● Have you checked your physical cabling or wireless network connectivity? ● Does your company/LAN have MAC filtering/port security on? You might need to modify switch settings for your drop, or talk to your network administrator. ● Have you checked your upstream gateway/SOHO router? Trying pinging the gateway’s IP address to see if your pfSense VM can reach the gateway. Try running the traceroute command to google.com to see if and where your ping command is |||||||||||||||||||| |||||||||||||||||||| failing. ● Do you have an upstream firewall in place? Is 443/TCP allowed outbound? Is 53/UDP allowed outbound? Is ICMP allowed? ● Did you input the commands correctly? Try entering them again. No, I’m not patronizing you. Once you have verified that your firewall’s connectivity, policies, and services are properly configured, SNAPSHOT THE VM BEFORE moving on and trying your hand at setting up the remaining VMs. 13.23 Your Turn The pfSense VM required a lot of custom configuration, both in the ESXi web interface, as well as in the VM itself once the OS was installed. Now that we did that together, I’m going to have you create the rest of the virtual machines yourself -- with the exception of the Metasploitable 2 VM since it’s a special case for reasons you will see momentarily. Below are a set of spec sheets to help you create the other virtual machines for the lab on your own. Think of them as checklists you should be able to run through unaccompanied. All of our Linux-based virtual machines are built on Debian Linux based distros (both, Ubuntu and Kali Linux are derived from Debian), so setup and configuration of all our lab VMs should be very similar. 13.24 Kali Linux VM Kali Linux is a popular penetration testing distro. Popular because hackers and script kiddies are lazy, and practically everything you need for performing a penetration test or joining anonymous is installed by default. Kali is going to be our loud, obnoxious attacker that we’re going to use as a noise generator for the express purpose of generating events on our IPS VM. I want you to perform the following tasks: ● Download Kali Linux 64-bit here: ● Upload your Kali Linux ISO file to your ESXi server’s datastore using the |||||||||||||||||||| |||||||||||||||||||| Datastore browser ● Create a VM with the following settings: ○ Name the VM “Kali” ○ Set the Guest OS family to Linux ○ Set the Guest OS version to Debian GNU/Linux 8 (64-bit) ○ Allocate 1 or 2 CPUs ○ Allocate 4GB of RAM ○ Connect the VM’s network adapter to the IPS 1 port group ○ Allocate 80 GB of space for Hard disk 1 ■ Configure the Disk Provisioning to Thick provisioned, lazily zeroed ■ Configure the Virtual Device Node to Sata Controller 0 (SATA (0:0)) ○ Remove the SCSI controller ○ Add a CD/DVD drive configured to use the Kali Linux ISO file you uploaded to the datastore ■ Ensure that the checkbox next to Connect at power on is checked ○ Adjust the Video Card setting Total Video Memory to at least 32MB ■ This is required in order for the VM to render the GUI ● Install Kali Linux ○ Note: If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter as the proxy server address. There is no username or password required to use the Squid proxy ● After the install is completed, power off the VM and adjust the following settings: ○ Remove the CD/DVD drive ○ Remove the USB controller ○ Under Network Adapter 1, make sure to document the MAC address of this VM ● Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP for the MAC address of adapter 1 for the Kali Linux VM; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to ● Power on the virtual machine and do the following: ○ Verify that the static mapping for the VM’s IP address worked; open a terminal/shell and run ifconfig -a | less and verify thateth0 (or whatever the |||||||||||||||||||| |||||||||||||||||||| network adapter’s name is in the VM) was given the IP address you configured its static mapping for in pfSense ■ If the virtual machine does not have an IP address, run the dhclient command to have the VM request an IP address from the DHCP server ○ Verify the virtual machine can reach the internet ■ In a terminal/shell run the commandcurl -I ● This will confirm DNS can resolve domain names, and that the VM can reach the internet ● If the command appears to hang or does not return output, use the Final Connectivity Checks and Troubleshooting guide we used for testing connectivity with pfSense. ○ Note: Theping command, in this case, will work from this virtual machine, due to the unique firewall policy on the OPT1 network ○ Update/patch the VM ■ In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait: ● export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y q dist-upgrade ○ Create a snapshot for the VM 13.25 SIEM VM SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for log aggregator; we will be having our IPS VM log its events here. We’re going to be running Splunk on this VM. Splunk is a commercial program for managing logs on a pretty large scale. By default, and with no licensing, Splunk only allows you to collect or index 500MB worth of logs per day, however, there are /ways/ around this that we’ll talk about later. I want you perform the following tasks to set up the SIEM VM: ● Download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: ● Upload the Ubuntu Server ISO file to your ESXi server’s datastore ● Create a VM with the following settings: |||||||||||||||||||| |||||||||||||||||||| ○ Name the VM “SIEM” ○ Set the Guest OS family to Linux ○ Set the Guest OS version to Ubuntu Linux (64-bit) ○ Allocate 1 or 2 CPUs ○ Allocate 4GB of RAM ○ Connect the VM’s network adapter to the Management port group ○ Allocate 80 GB of space for Hard disk 1 ■ Configure the Disk Provisioning to Thick provisioned, lazily zeroed ■ Configure the Virtual Device Node to Sata Controller 0 (SATA (0:0)) ○ Remove the SCSI controller ○ Add a CD/DVD drive configured to use the Ubuntu Linux ISO file you uploaded to the datastore. ■ Ensure that the checkbox next to Connect at power on is checked ● Install Ubuntu Server ○ Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You

connections to hosts on the Management network, or your physical, Bridged network. The next two rules explicitly deny HTTP and FTP traffic outbound. This is because the Squid proxy we will be installing can proxy both HTTP and FTP traffic. Due to our last firewall rule allowing any traffic in or out that isn't explicitly defined above, we have to block HTTP and FTP outbound to force systems on this network to use the Squid proxy, so that these connections can be logged. You may have noticed that I haven't explicitly blocked port 53/udp outbound for the IPS network. This is another malware analysis configuration I have accounted for. Some malware families will alter DNS settings, or use DNS tunneling for command and control, so I allow systems on this network to use external DNS servers. Please note, that if you are not planning on using the IPS network for malware analysis, you may want to adapt the firewall rules for the Management network 14.9 Network Configuration - Core Network Services As hinted above and throughout the sections dedicated to the setup of the different hypervisors, pfSense is configured to offer a few core network services for the systems of the lab environment. In the following sections, we will go over the services used, and how to set them up to meet the needs of our lab network. 14.10 NTP NTP is shorthand for Network Time Protocol. It is a service that allows systems to synchronize their internal clocks with a known (or at least assumed) good time source. Using NTP prevents system clocks from drifting, which has very important connotations for incident response, since you need to ensure that the timestamps on all the data collected are accurate. |||||||||||||||||||| |||||||||||||||||||| On the pfSense web UI, navigate to Services > NTP. In the listbox on top of the page, highlight the LAN and the OPT1 interfaces to ensure that upon completion, the NTP service and the adjusted settings will be made available for those two networks. (To select the two entries, press and hold the Ctrl key (Windows) or the Cmd key (macOS) respectively, click LAN and then, in turn, click OPT1.) By default, pfSense uses the NTP pool provided by its maintainers, and in most cases, this should work well enough. However, it never hurts to know alternatives. Visit the NTP Pool Project at and find the region of the world you are living in. In my case, I selected North America from the list of active servers, and on the next page, I was presented the following NTP server information: 0.north-america.pool.ntp.org 1.north-america.pool.ntp.org 2.north-america.pool.ntp.org 3.north-america.pool.ntp.org Using the information NTP Pool Project page for your region, you can click the Add button and add the additional NTP servers. After adding all four servers I found listed, I removed the default NTP server pool provided by pfSense by clicking on the corresponding Delete button on the right. After making the changes for the LAN and the OPT1 interface, click Save to enable the service. |||||||||||||||||||| |||||||||||||||||||| Note: If you are setting up your lab network on a corporate/enterprise network, you will need to ensure that port 123/udp is allowed outbound on your enterprise/local network. Otherwise, your network may be big enough to where there is already a designated NTP server in use on your network. You'll want to use the IP address of that NTP server instead, if necessary. 14.11 DHCP If you followed the instructions on how to configure the network interfaces of the pfSense VM in any of the previous hypervisor setup sections, then DHCP should be set up and enabled already. Systems on the network connected to the LAN interface should be served addresses from 172.16.1.10 to 172.16.1.254. Hosts on the OPT1 interface network should be served addresses from 172.16.2.10 to 172.16.2.254. While the CLI of pfSense lets you set the IP address scopes easily enough, the web interface allows a lot more granular control. DHCP configuration is available via the Services > DHCP Server |||||||||||||||||||| |||||||||||||||||||| page of the web UI. As you can see in the screen capture above, there are separate tabs for the LAN and OPT1 interfaces. Ensure the checkbox next to Enable DHCP Server on the [interface name] interface is checked (for both LAN and OPT1 interfaces), and confirm the corresponding IP address range for each interface is correct. By default, pfSense's DHCP server propagates the IP address set on the LAN and on the OPT1 interface as the default gateway as well as the DNS server for the corresponding network. For the purposes of our lab this is perfect, since pfSense handles the routing of the network traffic, has its DNS Resolver service enabled by default, and will use the DNS servers we configured during the initial setup for forwarding DNS requests. pfSense also allows us to configure static DHCP allocations for any network it is running DHCP for. Static mappings ensure that a virtual machine with a specific MAC address on its network card, connected to a particular network will ALWAYS get the same IP address from the DHCP server. The static mappings have to be for IP addresses from the network that the corresponding pfSense interface is in, but NOT for any address in the actual DHCP range. In the illustration above, the DHCP range is from 172.16.1.10 to 172.16.1.254. The addresses 172.16.1.1 and 172.16.1.2 are already in use by pfSense and our host system's network interfaces (if you are using a hosted hypervisor). That means we can only use the IP addresses from 172.16.1.3to 172.16.1.9 for static DHCP allocations, unless the DHCP range is adjusted. |||||||||||||||||||| |||||||||||||||||||| As an example, the following image shows the static DHCP mapping I am using for "Squadsight", the IPS VM's management interface, connected to the Management network. To create a static DHCP mapping, identify the MAC address of the network interface you want to create the mapping entry for, enter it into the MAC Address field (separated by colons [:]) choose the IP address you want the DHCP server to deliver that VM's network interface, enter it into the IP Address field and finally, save the settings. Note that the DNS Servers configuration is redundant, and you could get away with leaving the DNS Servers fields fields blank. 14.12 DNS Resolver By default, pfSense has the DNS Resolver service enabled. To verify this, navigate to Services > DNS Resolver. Verify that the Enable DNS resolver checkbox is checked, and that the All option is selected for Network Interfaces field as well as for Outgoing Network Interfaces field. |||||||||||||||||||| |||||||||||||||||||| To confirm which DNS servers pfSe nse will forward queries to if it can't resolve the domain name itself, navigate to System > General Setup, and view the DNS Server Settings. In my case, I'm using 8.8.8.8 (Google DNS), 4.2.2.2 (Level 3 DNS) and 192.168.1.1 (the default gateway/DNS for my physical network). 14.13 Squid Proxy |||||||||||||||||||| |||||||||||||||||||| pfSense supports utilizing the Squid HTTP proxy; all you need to do is install the package for it. Having a proxy server installed provides more granular control over the HTTP and FTP traffic allowed to flow into and out of the lab network. In addition to that, at a later date, we can choose to log the requests made, and have them sent to our SIEM VM to review event data. To install Squid, navigate to System > Package Manager, and click on Available Packages. In the search bar, type "squid", and click the Install button to the right of the squid package, then on the next page, click Confirm. This should kick off the installer. Upon completion, the installer log in the middle of the screen will read Success. To confirm that Squid was installed properly, navigate to Services. If Squid Proxy Server is listed as an option there, the installation was successful. Proceed by clicking on the menu entry. On the page that appears, click on the tab labeled Local Cache. We won't be modifying any of the default settings, however, pfSense requires you to confirm the default local cache settings before you can actually start the Squid service. Simply scroll to the bottom of the page, and click the Save button. |||||||||||||||||||| |||||||||||||||||||| Next, click on the General tab. Click to check the checkbox next to Enable Squid Proxy. In the Proxy Interface(s) pane, make sure that the entries for LAN and for OPT1 are highlighted; this ensures that the proxy server is available on both of these networks. Afterwards, scroll to the bottom of the page, and click Save. Note that with the firewall rules we have set up for the Management and the IPS network, we MUST use the proxy for all regular HTTP and FTP traffic. This makes using tools likewget or apt-get for setting up the other virtual machines a little bit more complicated. In order to work with command line applications that use HTTP or FTP for application updates in Linux/BSD environments, make sure that you familiarize yourself with thehttp_proxy variable and its use for Linux/BSD CLI tools. I found this great tutorial on how to set thehttp_proxy variable to ensure that this isn't a problem: |||||||||||||||||||| |||||||||||||||||||| . Note: The guide mentioned above assumes that the proxy is listening on port 8080/tcp, whereas our Squid proxy is listening on port 3128/tcp. Keep this in mind! 15 Defense in Depth for Windows Hosted Hypervisors Using either VMware Workstation on Windows, VirtualBox on Windows or Microsoft Client Hyper-V to set up your lab environment, you created at least one virtual network that added a virtual network adapter to the Windows host operating system. Each hypervisor has a different name for this network card: Client Hyper-V labels it vEthernet [vswitch name] (in our case, vEthernet (Management)) VirtualBox calls this interface VirtualBox Host Only-Network VMware Workstation identifies it as VMware Network Adaptervmnetx (where x is thevmnet or VM network the host is attached to; usually, the default hostonly network isvmnet1) If you are running your VM lab on a laptop or on a device that constantly moves from one network to another, you may not always be in control of your physical system's IP address for accessing your virtual assets. Adding a virtual adapter for the Management network (the LAN interface in pfSense) to your Windows hypervisor host allows it to directly interact with the IPS and SIEM VMs, without having to worry about firewall rules in pfSense, or adding multiple network routes on the host system. As previously stated, to reach the Kali VM on the IPS network, you need a firewall rule on the LAN interface to allow SSH access to the IP address of this virtual machine. In addition to that, you need to add at least one static network route on the Windows host in order to tell it where to send the packets to interact with the Kali host. While having this virtual network card directly attached to the Management virtual network is convenient, it also means that you are potentially exposing the host system to the devices on the Management network. Even with the pfSense firewall in place to enforce boundaries and segmentation between the various networks, taking additional security measures ensures we are not |||||||||||||||||||| |||||||||||||||||||| putting all of our eggs into one basket. The following sections will instruction you on how to unbind various Microsoft network protocols from the virtual network adapter, and how to implement a rule on the host's Windows Firewall that denies inbound connections on the virtual network adapter, but allows the virtual adapter to make connections to the virtual network as necessary. 15.1 Unbinding Network Protocols on Windows Virtual Adapters For the sake of simplicity, this tutorial is based on the VirtualBox Host-Only Network virtual network adapter. For the other hosted hypervisors discussed, you just need to substitute this description with vEthernet(Management) or VMware Network Adapter vmnet1 as necessary. If there are any special steps or considerations that should be noted for VMware or Client Hyper-V users, I will advise on differences as we come across them. To begin, navigate to the Network and Sharing Center, and click the Change adapter Settings option on the left of the screen. Alternatively you can head directly to View Network Connections. Find the adapter labeled VirtualBox Host-Only Network, right click it, and select Properties. On the Networking tab of the Properties window, under the pane entitled This connection uses the following items:, uncheck everything except for: ● VirtualBox NDIS6 Bridged Networking Driver ● QoS Packet Scheduler |||||||||||||||||||| |||||||||||||||||||| ● Internet Protocol Version 4 (TCP/IPv4) You may need to scroll down to disable all of the remaining items. Note for Client Hyper-V and VMware Workstation Hypervisors: Leave only QoS Packet Scheduler and Internet Protocol Version 4 (TCP/IPv4) enabled. Every other item listed here should be set to disabled. Next, click to select Internet Protocol Version 4 (TCP/IPv4), then click on the Properties button below the pane. First thing we need to do is to assign this network adapter a static IP address, so it can access resources in the host-only network, which represents our Management network. For my lab, I used the network 172.16.1.0/24, and configured the host-only adapter to have the IP address 172.16.1.2, the subnet mask of 255.255.255.0, and NO default gateway. You can input these settings after clicking the radio button next to Use the following IP address:. When finished, click the Use the following DNS servers: radio button, and leave the |||||||||||||||||||| |||||||||||||||||||| input boxes blank. Next, Click the Advanced... button. In the Advanced TCP/IP Settings window, switch to the WINS tab. At the bottom, in the section captioned NetBIOS setting, click the Disable NetBIOS over TCP/IP radio button. |||||||||||||||||||| |||||||||||||||||||| To finish, first, click OK to close the Advanced TCP/IP Settings window, next, click OK to close the Internet Protocol Version 4 (TCP/IPv4) Properties window, and finally, click OK to close the VirtualBox Host-Only Network Properties window. After you are done unbinding the nonessential network protocols from this virtual adapter, I would very highly suggest enabling and configuring Windows Firewall to deny inbound connections to this network card's IP address. 15.2 Using Windows Firewall to Limit Exposure of Windows Hypervisor Hosts As previously established, we are using the OS of the system the hosted hypervisor is installed on to manage our virtual machines. In my case, I'm |||||||||||||||||||| |||||||||||||||||||| using Windows as the operating system to host my hypervisor on. This section outlines how to create rules for the built-in firewall. In spite of whatever misgivings you might have, Windows Firewall is actually nice and easy to use; it serves our purposes perfectly. The aim of this guide is to specifically protect Windows systems hosting VMs via a hosted hypervisor that will be used to interact with systems in the virtual network. This guide will teach you how to create a firewall rule that will deny any and all traffic inbound towards the IP address we allocated to the virtual NIC (in our case, 172.16.1.2). While the pfSense firewall SHOULD be doing its job enforcing security boundaries, defense in depth means being paranoid. When set up, this Windows Firewall rule ensures that hosts on the virtual networks cannot initiate ANY connections to 172.16.1.2, but 172.16.1.2 is allowed to initiate and keep established connections to hosts as necessary. Let's begin by opening Windows Firewall with Advanced Security, selecting Inbound Rules, and then clicking on New Rule... to launch the New Inbound Rule Wizard. On the first page of the wizard, captioned Rule Type, select the Custom radio button, then click Next. On the next page, click the All Programs radio button, then click Next. |||||||||||||||||||| |||||||||||||||||||| On the following page, as we want this rule to apply to ANY protocol, accept the default settings by clicking Next. |||||||||||||||||||| |||||||||||||||||||| On the page captioned Scope, in the Which remote IP addresses does this apply to? section in lower part, ensure the Any IP address radio button is clicked. Above this, in the part headlined Which local IP addresses does this apply to?, select the radio button next to These IP addresses:, then click the Add... button on the right. |||||||||||||||||||| |||||||||||||||||||| We want this rule to apply to the address of the virtual network interface that is connected to the Management network. In our case, we have chosen is 172.16.1.2. Enter this value in the box below This IP address or subnet. Click OK to close the IP Address window, then click Next at the bottom of the New Inbound Rule Wizard window. |||||||||||||||||||| |||||||||||||||||||| On the Action page, click the Block the connection radio button, and again, click Next. |||||||||||||||||||| |||||||||||||||||||| Next up is the page with the available network profiles. Make sure that all three checkboxes (Domain, Private, Public) are all checked, then click Next. |||||||||||||||||||| |||||||||||||||||||| On the final page of the wizard, give your firewall rule a name, and add a description that reminds you why the rule exists, and click Finish. Make sure to document that this firewall rule exists. |||||||||||||||||||| |||||||||||||||||||| 16 Automated Patching for Linux Lab VMs When you created the lab VMs that are based on Ubuntu Linux, you were given the option to have the OS manage security updates for you. What if you want your Linux systems to automatically update ALL of their packages regularly? Proper maintenance and patching of your lab environment on a regular basis is VERY important. Because of the seriousness of this task, I created a simple script and process to follow in order to automate this task. 16.1 updater.sh Both Kali and Ubuntu Linux use the apt package manager to manage the software packages; this includes software updates. Below is a simple script |||||||||||||||||||| |||||||||||||||||||| that utilizes the apt-get command to check for, automatically download, and install the latest updates available, then reboot the system: #!/bin/bash #updater.sh - Weekly update script #This script checks for the latest updates, downloads them, then reboots the system. #Place this script in /etc/cron.weekly, ensure it is owned by root (chown root:root /etc/cron.weekly/updater.sh) #Ensure the script has at least user execute permissions (chmod 700 /etc/cron.weekly/updater.sh) #If you want updates to be ran once daily or weekly, simply place this script into /etc/cron.daily or /etc/cron.monthly as you see fit. export DEBIAN_FRONTEND=noninteractive apt-get -q update apt-get -y -q dist-upgrade logger updater.sh ran successfully. Rebooting system. init 6 exit 0 As the commented lines of this script read, place this script into/etc/cron.daily, weekly, or monthly, depending on whether you want your system to auto-update on a daily, weekly, or monthly basis. User root needs to be creator/owner of the script (chown root:root), and it must have at least the file permissions use and execute (chmod u+x, orchmod 700) in order to run successfully. When the script finishes, the system WILL be rebooted. For this reason, I would NOT, unless it has been agreed upon, deploy this script to a production system that runs anything of any importance. It is designed to keep your lab environment secured and up to date, and only your lab environment. The script will also write a line to/var/log/syslog via thelogger command, to let you know that and when it was run. You can use this script to keep the Kali, SIEM, and IPS VMs automatically |||||||||||||||||||| |||||||||||||||||||| updated, as well as any new VM you decide to add, as long as it uses .deb packages and the apt package manager. Hypothetically, the script can also be modified to use the equivalent update commands foryum (the package manager for redhat/RPM-based distros) or other package managers as well. 17 Remote Lab Management If you're interested in configuring remote management for your lab, read on. The following guides will assist you in setting up SSH access for your lab assets. There are sections dedicated to configuring remote access on Windows hosted hypervisors, on OS X/Unix/Linux hosted hypervisors, as well as on how to manage remote access to bare-metal hypervisors for either Windows or Unix/Linux users. 17.1 Windows Remote Access The following are a list of remote access solutions and related configurations aiming to allow users of a Windows hosted hypervisor to remotely administer and interact with their lab systems. 17.2 Persistent Static Routes Recall that all of our hosted hypervisors should have a virtual network card attached to the Management virtual network. This means that establishing network connections to the IPS and SIEM virtual machines should be easy, since the virtual network adapter and the two VMs are on the same local network. However, if we want our hypervisor host to communicate with hosts on the IPS network, such as the Kali VM, we need to tell our Windows host where to forward its traffic in order to reach the network (172.16.2.0/24) the Kali Linux VM resides in. Enter the route command. route allows you to edit the network routing table of the Windows host. We're going to create a persistent route to the 172.16.2.0 network. Navigate to the Command Prompt application in the Start Menu, or find it with the Windows Search. Right click on the icon/link for the program, and choose the option Run as administrator from the menu. |||||||||||||||||||| |||||||||||||||||||| In the Command Prompt session, execute the following command: route -p add 172.16.2.0 mask 255.255.255.0 172.16.1.1 This command instructs Windows to route network traffic destined for 172.16.2.0/24 (the IPS network) through 172.16.1.1 (which is pfSense's gateway interface for the Management network). The -p option makes this |||||||||||||||||||| |||||||||||||||||||| route persistent. By default, when the Windows system reboots, any routes added manually without that option are lost. If you want to confirm you entered the command correctly, run the following in the Command Prompt session: route print You'll notice that the output returned by the command has a section called Persistent Routes, and that the route we just added is listed there. You should be able now to access hosts on the 172.16.2.0 network. Please bear in mind, that while you may have a static route to the 172.16.2.0/24 network in the routing table, that in order to establish a connection to systems on that network, you must also have to have firewalls rules configured on pfSense VM to allow traffic from the Management network to the LAN network, and that the system you want to connect to must also be running the network service you want to interact with. 17.3 Windows SSH and SCP Software To access our lab VMs remotely, we will be using the SSH protocol. If you want to use Windows to manage your Linux systems over SSH, I recommend downloading and installing the following applications: ● PuTTYgen (sgtatham/putty/download.html) ○ PuTTYgen is a tool for generating SSH public and private keys for doing keybased SSH authentication from Windows SSH clients. ● mRemoteNG (○ mRemoteNG is a tremendously useful application for managing a wide variety of remote access protocols - RDP, SSH (v1 and v2), telnet, rlogin, Citrix ICA, etc. ● WinSCP (○ A graphical SCP client for performing SCP transactions from Windows clients. Also supports key-based authentication. Another tool that comes in handy now and then is a versatile text editor. While there is a wide range of programs in this category out there, I regularly use |||||||||||||||||||| |||||||||||||||||||| Notepad++ on Windows systems. So, in case you don't use it already or don't have another go-to editor by now, I suggest you get yourself a copy of the most current version at . 17.4 Generating an SSH key in Windows using PuTTYgen Doubleclick puttygen.exe to open the application. (Please note that if you're running Windows 8 or higher, you may get a Windows SmartScreen Windows Protected your PC prompt. Click More Info, then click the Run anyway button.) You are greeted with this screen: In the section headlined Parameters, make sure that the RSA radio button is selected. Next, in the input box to the right of Number of bits in a generated key:, change the default value to 4096. Finish by clicking the Generate button in the section labeled Actions. |||||||||||||||||||| |||||||||||||||||||| Once the system successfully generated both the SSH private and the public key, the section titled Key is updated, displaying details similar to those shown in the picture below: |||||||||||||||||||| |||||||||||||||||||| Feel free to modify the default key comment provided by PuTTYgen to something that makes more sense to you. In my case, I changed it to windowsmanagement to signify that this SSH key is for my Windows workstation. In addition to that, PuTTYgen offers the option to set a key passphrase for your new SSH key. If you decide to use one, this means that not only MUST you have the SSH key itself, and that you must ALSO know its passphrase in order to be able to use this SSH key to enable remote access to your lab VMs. You can also choose to leave the key passphrase field blank. Since this key is intended to be used only in a lab network environment that AT NO TIME should host anything of any value or significance, entering a passphrase isn't exactly important in this case; leave it blank. After adjusting the key comment, click the Save public key button. In the Explorer window that appears, enter a descriptive name likewindowsmanagement.pub for the public key file, and save it. |||||||||||||||||||| |||||||||||||||||||| Next, click on the Save private key button, and save the private key as well. I would highly suggest saving this file in the same place you stored the the public key. |||||||||||||||||||| |||||||||||||||||||| It is EXTREMELY important to keep these keys (especially the private one) in a safe location that only you have access to, in order to ensure others do not discover your SSH keys and attempt to abuse them. While this isn't much of a concern with a private lab environment, key security is something to ALWAYS be mindful of on real-world systems if you choose to use keybased SSH authentication. The next step involves making some minor edits to the public key we just saved, since Linux and BSD systems expect SSH public keys to be formatted in a very specific way. To do this, we will be using Notepad++. If you haven't done so already, download and install Notepad++ (. After doing so, open up Windows explorer and navigate to where you saved your public key file. Right click on the file, and select Edit with Notepad++. |||||||||||||||||||| |||||||||||||||||||| Notepad++ displays the contents of the file, which should look similar to what is shown in the picture below: As mentioned above, Linux and BSD systems expect SSH public keys to be |||||||||||||||||||| |||||||||||||||||||| formatted in a very specific way. The key must be a SINGLE continuous line, and no other information apart from the key itself is allowed. So first of all, remove the following lines: ● ---- BEGIN SSH2 PUBLIC KEY --● Comment: "windows-management" ● ---- END SSH2 PUBLIC KEY --What is left is the data the actual public key is made up of: Highlight all of the remaining text, then, on the main menu, go to Search > Replace... to open up the Find/Replace window with the Replace tab selected. In the Search Mode section at the bottom of the window, select the radio button next to Extended. Next, make sure that the Replace with: input box is empty, then enter the following for Find what: field: |||||||||||||||||||| |||||||||||||||||||| Click Replace All, then click Close to exit the Find/Replace window. We just replaced all of the newline and carriage return characters in our text file, turning the key into a single, continuous line. Now, at the beginning of the line, insert the text "ssh-rsa". It is important to include the blank space at the end, exactly like it is given in the quotation marks. Next, at the end of the line, add the text " windows-management" (or whatever comment you want to associate with this key). This time around, it is mandatory to have the blank space between the END of the key and the beginning of the comment. If done correctly, the key should look like this: Note: that the line wrap after ssh-rsa is caused by the trailing blank space; there is no actual carriage return. After modifying the text as described above, select Edit > EOL Conversion > Unix (LF) from the notepad++ main menu. |||||||||||||||||||| |||||||||||||||||||| This option converts newline characters from the windows format or \r (the symbols that represent carriage return and newlines in a Windows text file) to just (newline characters). While the file should only be a single line, I have found that Unix/Linux systems behave strangely if this EOL Conversion has not been performed in advance. Afterwards, select File > Save As... from the main menu. In the Save As dialog window, change the value in the File name: input field to windowsmanagement-formatted.pub. Make sure that the file is going to be stored in the same folder the public and private SSH keys generated with PuTTYgen have been placed in. Click the Save button, and finish by quitting Notepad++. |||||||||||||||||||| |||||||||||||||||||| 17.5 Using mRemoteNG - Connection Files Start up mRemoteNG and open the View menu. From the highlighted options, click on Sessions, Notifications and Quick Connect Toolbar. Removing these items from view declutters mRemoteNG's user interface significantly. |||||||||||||||||||| |||||||||||||||||||| Disabling the above-mentioned views leaves us with three panes in the main screen: Connections, Config, and a large, grey window. In the menu bar on top of the Connections pane, first, click to highlight the Connections root entry. Next, click on the folder icon to add a folder underneath the Connections root. Name the new folder Lab Network. Click the Lab Network folder to highlight it, then click the leftmost icon in the Connections pane menu bar . This creates a new connection entry underneath the Lab Network folder. In the Config pane, by default placed right below the Connections pane, a set of numerous options for the newly added entry is displayed. We will go through the necessary settings for creating this first SSH connection (the SIEM VM) together. Afterwards, you will create the remaining two connections (for the Kali and IPS VMs, respectively) on your own. |||||||||||||||||||| |||||||||||||||||||| Starting in the Display section first, in the row labeled Name, double click on New Connection and insert "siem" in the box. Next, in the Protocol section, select RDP in the Protocol field. Click the drop-down arrow that appears, then select SSH version 2 from the corresponding list. After that, in the Connection section, double click on the input field next to Hostname/IP, and enter the value "172.16.1.3". Then double click on the input field in the row named Username, and insert the username you used during the Ubuntu 16.04 installation of the SIEM VM (in my case, I named my first userayy). Finally, double click on the input box to the right of Password to provide the password corresponding to the previously entered username. When you're all done, double click on siem in the Connections pane. If the username and password were entered correctly, you should automatically authenticate and log in to the SIEM VM with no problems. |||||||||||||||||||| |||||||||||||||||||| Continuing the above example, it's now your turn. Create two more connections under the Lab Network folder, using the following settings: ● Name: ips ● Protocol: SSH version 2 ● Hostname/IP: 172.16.1.4 ● Username: ● Password: ● Name: kali ● Protocol: SSH version 2 ● Hostname/IP: 172.16.2.2 ● Username: root ● Password: ● Note: You will need to enable SSH on Kali Linux before you attempt to connect. If you configured everything correctly, the connections panel should host two new connections: Their individual config options should look like this: |||||||||||||||||||| |||||||||||||||||||| Feel free to test SSH connection to the IPS VM, which should already work just fine, but do note that for the time being, you will NOT be able to reach the Kali VM (Yes, even if you added a static route for the 172.16.2.0/24 network). Don't worry about this for right now, we're going to fix this shortly. 17.6 Using mRemoteNG - PuTTY Saved Sessions Now that we have created the necessary SSH connections in mRemoteNG, the next step is to set up what is called a session profile. This will allow us to use the SSH key that we generated with PuTTYgen to authenticate to our lab VMs, instead of using a password. Or, if you password protected your private key earlier, it sets up a rudimentary form of two-factor authentication (e.g. you need to have the SSH key, and you must know the SSH key's password in order to connect). From the main menu of mRemoteNG, choose Tools > Options. In the Options window that appears, select Advanced in the left pane, then click the Launch PuTTY button. This will open a window captioned PuTTYNG Configuration. On the left pane labeled Category, click on the + symbol next to SSH to expand the menu branch, then click on the Auth entry (this time the text itself, not the leading + sign). |||||||||||||||||||| |||||||||||||||||||| In the section headlined Authentication parameters, click the Browse... button next to Private key file for authentication:. An Explorer window pops up. Navigate to the folder where you stored the PRIVATE SSH key (the file with the.ppk extension) generated with PuTTYgen earlier, then double click to select it. |||||||||||||||||||| |||||||||||||||||||| After selecting the file that contains the private SSH key, click on Session under the Category: pane. In the section titled Load, save, or delete a stored session, insert "Lab_Network_SSH" into the Saved Sessions input box, then click the Save button. |||||||||||||||||||| |||||||||||||||||||| Click the Close button to exit PuTTYNG Configuration, then click the OK button in the mRemoteNG Options window. To return to the main screen. Select the entry for siem in the Connections overview to bring up the corresponding settings in the Config pane. Under the Protocol heading, select the entry named PuTTY Session. Click on the drop-down arrow on the right and pick Lab_Network_SSH from the list. |||||||||||||||||||| |||||||||||||||||||| Return to the Connections pane, and double click on the siem entry to start an SSH connection to the respective VM. In the terminal window that appears, notice the text stating that the Server refused our key, yet the login attempt still succeeds. mRemoteNG establishes an SSH connection and to authenticate with the server using the SSH key we added to our PuTTY session. The problem is that the user attempting to log in doesn't have the public key for the used private key in itsauthorized_keys list on the remote server. So the server responds, "The user doesn't have the corresponding public key in their authorized_keysfile." It immediately falls back to password-based authentication, accepting the password provided by mRemoteNG, and the user is logged in successfully. We'll fixing this problem in the following section. |||||||||||||||||||| |||||||||||||||||||| 17.7 Enabling Key-Based Authentication on Linux/Unix systems If you are continuing on from the previous section, you might still have an open SSH session to the SIEM VM, logged in with the user created during the initial system setup. In that case, you should already be in home directory of the connected user. If you are not, or if you just want to make sure that you are, run the commandcd ~/ inside the current SSH session. If you are not or no longer connected to the SIEM VM, open up mRemoteNG and create a new session. Once you are logged in, insert and execute the following: mkdir ~/.ssh;chmod 700 ~/.ssh; touch ~/.ssh/authorized_keys;chmod 600 ~/.ssh/authorized_keys To verify everything has been processed properly, run the following command: ls -al ~/.ssh The output displayed should look similar to the one shown in the picture below: The first command was actually made up of four instructions in rapid succession. It created the hidden directory.ssh, gave this directory file permissions that grant access only to the user who created it, then generated the empty fileauthorized_keys with thetouch command (if the file does NOT already exist, touch creates it), and finally changed the permissions of this file to only allow the account currently used has read and write access. The next part is tricky, in that we have to insert the contents of windowsmanagementformatted.pub on our local machine into ~/.ssh/authorized_keysin the user's home directory on the remote system. There are a several ways to do this. In the following sections, you will find three different approches 17.8 Key Copy Method 1: echo append to authorized_keys Open up the file windows-management-formatted.pub in Notepad++, highlight all of the text in the file, right click on it, then select Copy. The contents of the public key are now copied to the Windows clipboard. |||||||||||||||||||| |||||||||||||||||||| Back in mRemoteNG, in the open SSH session to the SIEM VM, enter the following command: echo '[right click here]' >> .ssh/authorized_keys Be sure to include the single quotes in the command above! By right-clicking inside the current SSH session (make sure to right click only once), the contents of the Windows clipboard were pasted into the terminal window. The text from the windows-management-formatted.pub file should now appear in between the single quotes. The final portion of the command in the terminal instructs the system to append this (echoed) key data to the.ssh/authorized_keys file. If your (somewhat extended) command line looks similar to the one you can see in the picture below, execute it. The cat command reads the contents of a file and writes it to the terminal. Your output should look similar to this: 17.9 Key Copy Method 2: usingvi |||||||||||||||||||| |||||||||||||||||||| The vi text editor is older than dirt. It's hard to use, hard to figure out, and has arcane invocations that nobody understands. And now you get to struggle with it as your elders did before you. Think of it as a rite of passage in the Unix/Linux world. Open up the file windows-management-formatted.pub in Notepad++, highlight all of the text in the file, right click on it, then select Copy. The contents of the public key are now copied to the Windows clipboard. Return to the open SSH session to the SIEM VM in mRemoteNG, and enter the following command: vi ~/.ssh/authorized_keys This opens the vi editor in review mode. The authorized_keys file should be blank. No text, no other keys, nothing. Hit theikey to enter vi's insert mode, then right click on the terminal window JUST ONCE. This fills in the contents of the Windows clipboard into the current editor session. After the contents have been copied into the open document, hit the esc key to exit the insert mode, then type: :wq! to save the file and exit vi. Thecat command reads the contents of a file and writes it to the terminal. Your output should look similar to this: 17.10 Key Copy Method 3: SCP If you haven't done so yet, download the current version of WinSCP from |||||||||||||||||||| |||||||||||||||||||| then install it on your Windows machine. Open the program, and select New Site in the left panel. On the right, in the section captioned Session, input the host information for the SIEM VM, along with the corresponding username and password,,then click the Login button. Once logged in successfully, you are greeted with with a screen made up of two panes: the left pane represents files and locations on the local computer, the right one those on the remote computer. Double click on the directory path on top of the left pane (the portion in blue, right below the menu toolbars). In the Open directory window that pops up, click the Browse button. Navigate to the directory that holds the SSH public and private keys, then click OK. |||||||||||||||||||| |||||||||||||||||||| The left pane of the WinSCP window changes to display the contents of the selected directory. Click on the file windows-management-formatted.pub, then drag it from the left pane to the right one. The file should appear in the right pane. Exit WinSCP, and start mRemoteNG if it isn't already running. In case the previously used session is closed, open an SSH connection to the SIEM VM. Once connected, run the command: cat windows-management-formatted.pub If the output in the terminal window resembles the one shown in the image above, the file made it over in one piece. Next, enter and execute the following: |||||||||||||||||||| |||||||||||||||||||| cat windows-management-formatted.pub >> ~/.ssh/authorized_keys Next, run the following command: cat .ssh/authorized_keys Thecat command reads the contents of a file and writes it to the terminal. Your output should look similar to this: 17.11 Making sure it worked After verifying that the public key has been successfully copied to .ssh/authorized_keys, run exit inside the terminal window to leave the SSH session and log off of the SIEM VM. In mRemoteNG, on the Config pane, click on the Password entry for the siem connection. Clear the corresponding input field Then double click on siem in the Connections pane to reconnect to the VM. If the setup of the private SSH key for the user is correct, and if you didn't put a passphrase on the SSH key we generated with PuTTYgen, you should be greeted with a login prompt with no need to enter a password. The server is |||||||||||||||||||| |||||||||||||||||||| comparing the public and private key to authenticate the user to the SIEM VM. If the comparison is successful (e.g. the public key on the system is related to the private key that has been sent to authenticate), the user will be logged in. If you DID set a key passphrase during generation of the SSH keys earlier, you are prompted to enter the password for that key now. Once you have verified that key-based authentication is working on the SIEM virtual machine, you can repeat the process to enable key-based auth on the IPS VM. You may also wish to jump to the section How to Enable SSH on Kali Linux, if you planned on enabling remote access for the Kali LInux virtual machine. 17.12 How to use Key-Based Authentication with WinSCP Assuming that you followed the instructions in the sections above, and furthermore assuming that you're using a Windows system, you have configured (or should be able to configure) keybased authentication for SSH. Another common usecase for this kind of authentication is SCP. SCP is a protocol that relies on using SSH as a transport which makes it a (more) secure alternative to FTP. In fact, one of the abovementioned methods for adding your SSH public key to the Linux servers in our lab makes use of SCP. Just like with the key-based authentication guides above, I will instruct you on how to enable this functionality on the SIEM virtual machine, and leave it as an exercise to you, if you would like to enable this functionality on the remaining lab VMs. |||||||||||||||||||| |||||||||||||||||||| First, open WinSCP. You are greeted with a window captioned Login. The pane on the left shows a highlighted entry with a computer monitor icon labeled New Site, on the right side of the window is a section headlined Session. As we are creating a session for the SIEM VM, enter 172.16.1.3 into the input box below Host name: . Leave the default value of 22 for the Port number. Into the field titled User name:, insert the name of the account you configured SSH keybased authentication for on the SIEM VM (in my case, the user's name was ayy). You may have noticed that I haven't mentioned the Password: field in WinSCP yet. That's because we aren't going to use it. Instead, after entering the name of the user you enabled keybased authentication for, click on the Advanced... button. The Advanced Site Settings window pops up. On the left pane, in the section titled SSH, select the entry named Authentication. In the middle of the window, find the Authentication parameters section. On the right side of the input box below Private key file:, click on the gray button with the ellipses in it. In the Explorer window that appears, navigate to the folder containing the PRIVATE SSH key that |||||||||||||||||||| |||||||||||||||||||| corresponds to the public key already set up on the host. Select this private key file to return to the Advanced Site Settings window, then click OK to close it. Back in the Login window, click the Save button. In the window captioned Save session as site that appears, change the suggested value in the box underneath Site name: to "[username]@siem". Using this pattern for the site names can make it easier to differentiate which session maps to which lab virtual machine. When you are done, click OK to finish. |||||||||||||||||||| |||||||||||||||||||| A new saved session appears on the left pane in the Login window. Double click on this new entry. If everything works as expected, an SCP session to the SIEM VM is established. 17.13 Linux, BSD, and OS X Remote Access This guide is dedicated to those running a hosted hypervisor on Linux, BSD, or OS X, or planning to use a workstation running one of the aforementioned operating systems to establish network access to the lab environment as required (e.g. for both, bare-metal and/or hosted hypervisor networking). The following sections include instructions on how to configure static routes and how to make them persist between reboots, on applications and commands I recommend to make remote management even easier, and finally on how to generate and use SSH keys for fast, easy and secure connectivity to the Linux lab VMs. 17.14 Static Routes in Linux and OS X Almost all Linux and BSD systems (yes, that includes OS X) come with the route command pre-installed for managing the system's network routing table. However, the syntax differs slightly between Linux and BSD variants. Additionally, Linux distros in general are deprecating theroute command in favor of the ip command, specificallyip route add. For this reason, we'll dis cuss using the ip command for adding static routes on Linux systems, while the guide for BSD and OS X on this matter is going to focus on using the respective version of the route command. |||||||||||||||||||| |||||||||||||||||||| 17.15 Adding Routes to Linux with the ip Command The ip command is considered the preferred way to request network information on modern Linux systems, as well as add new routes to the system's network routing table. Entering the command: ip route show displays the current contents of the OS' routing table. The output in the image above indicates that the default gateway is 172.16.1.1 (the IP address of the LAN interface of the pfSense VM, connected to the Management network). The default gateway is the address a host will forward any traffic to that isn't sent to target machines on its local network, in hopes that the gateway knows where to relay this data to. Another thing that can be learned from the output is that the machine is connected directly to the 172.16.1.0/24 network through itseth0 interface. Let's assume we are using a Linux workstation as our hypervisor host, and we want to access the 172.16.2.0/24 network (the IPS network) from our virtual network adapter connected to the 172.16.1.0/24 network (the Management network). To update the system's routing informations accordingly and verify the configuration, enter the following commands: ip route add 172.16.2.0/24 via 172.16.1.1 ip route show As you can see in the image above, the output produced by ip route show includes 172.16.2.0/24 via 172.16.1.1 dev eth0, indicating that the routing information has been added successfully. The device name (eth0) may differ, depending on the version and distribution of Linux being run, but other than that, the output should be consistent. Please bear in mind, that while you may have a static route to the 172.16.2.0/24

~/.ssh/authorized_keys; wc -l ~/.ssh/authorized_keys The cat command reads the contents of a file back out to the terminal, while thewc command counts the words of in file. In our case, the-l option is used to count the number of lines, because is VERY important that the copied content ofid_rsa.pub got written to the authorized_keys file as one single line. A public key entered into the authorized_keys file mustn’t have any newline characters. If there are any, they need to be removed. Each key entered into authorized_keys should only register as a single line in the file, and considering this is the first public key that has been inserted, it is expected to be the ONLY line in the file. Your output should look similar to this: 17.29 Key Copy Method 2: usingvi The vi text editor that is older than dirt, hard to use, hard to figure out, and has arcane invocations that nobody understands. And now you get to struggle with it as your elders did before you. Think of it as a rite of passage in the Unix/Linux world. Similar to what was outlined in method one, for the following approach, you will either need to open id_rsa.pub in a text editor of your choosing, or use thecat command to have the content of the file displayed in the terminal. Again, highlight all of the text the file contains, and copy it to your clipboard via ctrl + c or meta + c. |||||||||||||||||||| |||||||||||||||||||| Back in the SSH session to the IPS VM, enter the following command: vi ~/.ssh/authorized_keys This will open the editor in review mode. There should be nothing in the file. No text, no other keys, nothing. Press the ikey to enter vi’s ‘insert mode’. Hit Ctrl+V (Command+V in OS X) to paste the contents of the clipboard into the current editor session. After the contents have been copied into the open document, hit the ‘esc’ key to exit the ‘insert mode’, then type in :wq! to save the file and exit vi. |||||||||||||||||||| |||||||||||||||||||| As the finishing step, run the following command: cat ~/.ssh/authorized_keys; wc -l ~/.ssh/authorized_keys As outlined at the end of method 1 above, these commands help to ensure that the key got copied correctly, and there is only a single line in the authorized_keys file. 17.30 Key Copy Method 3: SCP Method 3 involves using the scp application to upload a copy ofid_rsa.pub to |||||||||||||||||||| |||||||||||||||||||| the IPS VM. From there, we can take the contents of said file and simply append them to the existing authorized_keys file. On your Linux, Unix, or OS X host, run the following command: scp ~/.ssh/id_rsa.pub :~/id_rsa.pub Replace ayy in the instruction above with the name of the user you set up on the IPS VM. This commands tells scp to to copy theid_rsa.pub from our local system, and put it in the home directory of userayy on the remote system. Next, if you don’t already or still have an open SSH session on the IPS VM, establish a connection as the user you set up on that host, then run the command ls -al ~/ to verify that there is a copy of theid_rsa.pubfile in the user’s home directory. From here, enter and execute the following: cat ~/id_rsa.pub >> ~/.ssh/authorized_keys Slightly deviating from how it has been used before, this time, catdoesn’t read |||||||||||||||||||| |||||||||||||||||||| back the content ofid_rsa.pub to the terminal. Instead, the output of the utility is directly appended to the authorized_keys file. Next, run the following command: cat ~/.ssh/authorized_keys; wc -l ~/.ssh/authorized_keys Just like with the other methods, this verifies that the key has been copied correctly and is intact. If the key is not on a single line or appears to be malformed, you will need to correct the errors. After verifying that the key has been added to theauthorized_keys file correctly, I suggest that you run the command rm -rf ~/id_rsa.pub the IPS VM to remove the uploaded public key. If you like to, you can keep the file on the system until after testing that key-based authentication is working as intended. 17.31 Making Sure it worked With the public key copied to the IPS VM, we have to ensure that the data was copied correctly, and that there is no error regarding the files needed. To test the setup, disconnect any SSH sessions established with the virtual machine, and simply try to reconnect to the IPS VM with the same user account as before. At this point, you should be able to achieve this by running ips If your host system recognizes this alias, an SSH session to the IPS VM should |||||||||||||||||||| |||||||||||||||||||| be opened. If you haven’t set up aliases yet, or don’t want to bother right now, simply run ssh If everything was copied correctly and key-based auth was configured correctly, you should have a session on the IPS VM without ever having to enter a password. (That is, unless you configured a password for your SSH key. In that case,the SSH session WILL ask for theid_rsa password while establishing the connection.) Note for OS X users: If you did set up a key for your id_rsa, and if you have ever entered it previously, this password has been added to your system’s keyring. (Without going into too many details, the Keychain is the default OS X credential manager.) If you want to see whether or not OS X is taking your id_rsa (SSH key) password from the keyring, or if you are trying to troubleshoot issues with your key-based authentication failing, try this out: Run the command ssh -vv (or you can use yourips alias and simply run “ips -vv”) When the option -vv is added, the ssh command is started in very verbose mode, showing a lot of what is going on in the background. Very verbose mode can generate a lot of output. The picture below shows an extract of what a successful key-based authentication looks like: |||||||||||||||||||| |||||||||||||||||||| You can see in the output above that our SSH key is sent to the remote system, and that the server accepts it. Note the line using passphrase from keychain appearing immediately after that. This is the OS X Keychain that I referenced above: when I used it before, the system saved the password for my SSH key, and applied it now automatically. If you get anything other than Authentication succeeded (publickey). in your verbose SSH output, or if you are prompted for the user’s password, the attempted key-based authentication for SSH has failed, and you need to do some troubleshooting. In order to help you doing so, here are some things you might want to consider: Did you copy ALL the contents of the id_rsa.pub file? Did you copy the contents of theid_rsa.pub file to~/.ssh/authorized_keys? Did you ensure that file permissions for the ~/.ssh directory were set to 700, and those for authorized _keys to 600 via the chmod command? (Note that many versions of Linux will REFUSE to use key-based authentication if the.ssh directory and files therein are NOT secured properly.) When you copied id_rsa.pub to theauthorized_keys file, were the entire contents of id_rsa.pub placed on a single line? When you configured your alias, or ran thessh command to connect to the IPS VM (172.16.1.4), did you specify the correct username (e.g.ssh , if your user’s name isayy)? 17.32 Using key-based authentication with the SCP command If you configured key-based authentication for SSH for a certain user, and you attempt to SCP to or from a system in which you enabled key-based authentication for said account, then the scp utility should automatically be able to utilize the SSH key. To demonstrate, I created the file foo.txt on the IPS VM, that simply contained the text, “This is a test.” Next, I ran the command: |||||||||||||||||||| |||||||||||||||||||| scp :~/foo.txt ~/foo.txt The program copied the file effortlessly. The SSH key also allows you to upload files from your local system to the IPS VM just as effortlessly. On my OS X system, I created the file bar.txt, that contained the text, “This is also a test.” Next, I ran the command: scp ~/bar.txt :~/bar.txt Once again, the file was copied without any hassle, as the SSH key had been configured properly for the corresponding user on the target host. |||||||||||||||||||| |||||||||||||||||||| 17.33 How to Enable SSH on Kali Linux For one reason or another, Kali Linux installions, by default, come with the SSH service disabled by default. It is believed that this is by design, in order to reduce exposure and possible attack surface for a system with a vast array of offensive security tools. Not to mention the fact, the only and default user for Kali is root; it is generally frowned upon to SSH in with this account. For the sake of usability, and because this is meant to be applied only in our lab environment, we are going to enable SSH on this virtual machine anyway. Power on your Kali Linux VM, and log in. Open up theTerminal application, then enter and execute: service ssh status From the status information that is returned, which should be similar to the one |||||||||||||||||||| |||||||||||||||||||| in the illustration above, you can tell that the SSH service is indeed disabled. To change this, run the following command: systemctl enable ssh This commands tells systemd, the system services manager, that we want the SSH service to start on boot. Next, we need to make a temporary change to the file /etc/ssh/sshd_config. Before we do that, we are creating a backup of this file by running the following command: cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak With the backup copy created, it is safe to edit the sshd_config file. Open /etc/ssh/sshd_config with your favorite editor (Kali Linux includes Leafpad, a graphical editor, if you don’t want to struggle with vi). Find the line PermitRootLogin prohibitpassword. Replace the portionprohibit-password withyes. This will allow the root user to authenticate over SSH with just a password. This is meant to be |||||||||||||||||||| |||||||||||||||||||| TEMPORARY. Save the file, and exit the editor. For the changes to take effect, we need to reboot the system. You can use the hypervisor’s power controls to reset the VM, or restart the virtual machine from inside the OS by whatever means you’re most comfortable (e.g. init 6, reboot, the shutdown button on the desktop, etc.). While the Kali Linux system is rebooting, if you’re using a Windows machine as your management or hypervisor host system, open mRemoteNG. If you haven’t done so while working through the corresponding section of this guide, create a new SSH version 2 connection to 172.16.2.2, using the PuTTY profile we set up earlier to enable key-based authentication. When the VM is up and running again, double click on the new Connections item to open an SSH session to the Kali Linux system. (For the moment, it is safe to ignore the message that the server refused the key. We are going to fix this in an instant. Right now, just use the password for the userroot to log in.) If you’re using Linux or OS X, just wait for the VM to finish booting, then open a new terminal window or tab, and run |||||||||||||||||||| |||||||||||||||||||| ssh When prompted for it, enter therootuser’s password to get a session on the Kali VM. If you weren’t able to establish an SSH connection, check the LAN network firewall policy on the pfSense firewall to make sure that traffic on port 22 is allowed from 172.16.1.1 to 172.16.2.2. Furthermore, check your system’s routing tables and ensure that the network 172.16.2.0 can be reached. Now that you have an SSH session on the Kali VM, depending on whether your hypervisor host or workstation of choice is running Windows, Linux, Unix, or OS X, you’d want to follow the directions for enabling key-based authentication for your Windows hypervisor host/workstation or for enabling key-based authentication for your Linux, Unix, or OS X hypervisor host/workstation. Copy your SSH public key to the ~/.ssh/authorized_keys on the Kali VM using any method you see fit. Afterwards, verify that key-based authentication is working for the Kali VM. Using mRemoteNG on Windows, this is as simple as disconnecting the SSH session, removing theroot user’s password from the Config details of the corresponding Connections item, and attempting to connect again. In the case of Linux, Unix, or OS X systems, disconnect your current SSH session, then reconnect using the command ssh If you get a session without having to enter root’s password (again, having to enter a password forid_rsa is expected, if you did set one), everything is working perfectly. Now that using a password for userroot is no longer needed in order to log in to the Kali VM via SSH, we can and should revert the sshd_config file to its original state: cp /etc/ssh/sshd_config.bak /etc/ssh/sshd_config At this point, you should be able to SSH in to the Kali system asroot, and password authentication for this account should be prohibited by the SSH service on that virtual machine. 17.34 Enabling, and securing root SSH |||||||||||||||||||| |||||||||||||||||||| Depending on who you talk to, people believe that you should never log in to a system using the root account, and thatroot account access over SSH should never ever happen. This is (most certainly) why the root account is disabled by default on the Ubuntu Linux distribution, and why it, just like many other current Linux distros, requires you to create a user level account as part of the installation. On real-world systems, theroot user account should be used sparingly, and SSH for root should never be enabled, unless absolutely necessary. However, this is a lab environment, OUR lab environment, we’re the only user here, the lab network has a VERY strict access control to limit our risks, and we take snapshots of our VMs. I argue, that for a personal lab environment, SSH asroot, with proper security in place, is perfectly fine. Just be careful, and know the risks. This section is entirely optional, and is intended for the extremely lazy only. I’m going to show you how to enable SSH for root on the SIEM VM (running Ubuntu). After that, you will learn how to disable password authentication for SSH entirely, for bothroot and non-root users. From there, you can decide if you want to apply these settings to your other Linux VMs. Completely disabling password authentication for SSH removes any possibility of brute force attacks, as the SSH server would drop the connection if an attacker were to try to use a password for authentication. Note: If you completed the section How to Enable SSH on Kali Linux, then you’re already using SSH for theroot user on that system, and you should have already configured Kali to use keybased authentication for this account. If you are planning to keep up that setup, I very highly recommend reading on, and disabling password authentication over SSH on the Kali VM, too. 17.35 Adding your SSH public key to root’sauthorized_keys file Using either mRemoteNG on Windows, or the ssh command on Linux, Unix, or OS X systems, establish an SSH connection to the SIEM VM as the user you created during the initial setup of that system (in my case, I named the accountayy), and run the following command to gain access to a root shell: sudo su |||||||||||||||||||| |||||||||||||||||||| sudo is a special command on Unix/Linux systems. If a user account exists in the /etc/sudoers file, it is allowed to run certain commands with root privileges, given that this user’s password can be provided. Since the first user you create during the setup of Ubuntu is in /etc/sudoers, and as there are no restrictions regarding the commands this user is allowed to execute via sudo, the command above gives us root shell access immediately. Normally, you need the root user’s password to do this, but since we ransudo prior to su, the system views our user account already as root, and thus just spawns a root shell for us. This allows us to login as root, even on Ubuntu, where theroot account is restricted. We now have to add our SSH public key to/root/.ssh/authorized_keys. Depending on whether your hypervisor host or workstation of choice is running Windows, Linux, Unix, or OS X, you’d want to follow the directors for enabling key-based authentication for your Windows hypervisor host/workstation, or for enabling key-based authentication for your Linux, Unix, or OS X hypervisor host/workstation. This step usually boils down to creating the.ssh folder in the home directory of the userroot, and adding our SSH public key to /root/.ssh/authorized_keys, while ensuring that file permissions for both the .ssh directory and authorized_keys file are set to only allow theroot user to access them. After adding the SSH public key to the rootuser’sauthorized_keys through the means you are most comfortable with, if you are on Windows, open mRemoteNG and add a new connection, using the following settings: |||||||||||||||||||| |||||||||||||||||||| When finished, make sure that all connections to the SIEM VM are closed, then double click on the new Connections item to open an SSH session to the virtual machine. If everything went well, you should be logged in as userroot. In case you are on a Linux, Unix, or OS X system, after updating theauthorized_keys file on the remote system, exit all connections to the SIEM VM, then run the command ssh to verify that your SSH key setup to the SIEM virtual machine is functioning properly. You can repeat this process on the IPS, and Kali Linux VMs as well. Note: If you enable key-based authentication as the root user for any or all of your VMs, I very highly recommend completing the next section. Before doing so however, ensure that all of your SSH keys are working correctly, and that you have easy access to them before continuing with the next section. Note: If you are using Linux, BSD, or OS X, you may want to review the section on using the alias command in order to add new or modify existing aliases in your local user’s .bashrc or .bash_profile, to simplify opening SSH connections to the SIEM VM as userroot. You can either use your favorite text editor to change (or create) any of said files, or you can simply create and append a new, unique alias to one of them by running a command similar to the one displayed in the picture below. Make sure that it is apparent that this alias logs you in to the SIEM VM as user root. |||||||||||||||||||| |||||||||||||||||||| 17.36 Disabling password authentication entirely via sshd_config Note: Before proceeding with editing the sshd_config file on any of your VMs, you will want to make a backup of thesshd_config file to revert to in case one of the modifications goes wrong. Run the command cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak to make a backup copy. If you run into any problems, you can restore the backup by running the command cp /etc/ssh/sshd_config.bak /etc/ssh/sshd_config While the local login for the user root is disabled in Ubuntu Linux, its SSH server (sshd) is set to allow connections for this account, as long as key-based authentication is used for the attempted connection (and the correspondingauthorized_keys file has been set up accordingly). This is because of thePermitRootLogin without-password directive in the /etc/ssh/sshd_config. If for some reasonPermitRootLogin is set to anything other than without-password, orprohibit-password, modify the directive to either of those settings. (When used with said directive, the values withoutpassword andprohibit-password can be used interchangeably. You may encounter one or the other, depending on the Linux distro used.) |||||||||||||||||||| |||||||||||||||||||| Additionally, we need to edit thePasswordAuthentication directive in /etc/ssh/sshd_config, setting its value fromyes to no. Change respective line in the file from to Remember this in the event you ever have a need to revert these changes. Save the /etc/ssh/sshd_config file, then restart the SSH service through the command: service ssh restart This change of setting disabled the use of passwords to authenticate SSH sessions entirely. As a consequence, the only means left for connecting to the remote system over SSH is key-based authentication. This affects all other user accounts on the customized system as well! |||||||||||||||||||| |||||||||||||||||||| In the illustration above, I attempted to log in to the SIEM VM with the account of user ‘ayy’, using the default PuTTY session instead of key-based authentication, and the connection was dropped immediately. Disabling the use of passwords to authenticate SSH sessions provides an extra layer of security, but also means you must keep control of your SSH private and public keys, ensuring they remain secure. If you lose the keys, the only way left to access systems adjusted as described above is through the hypervisor’s |||||||||||||||||||| |||||||||||||||||||| console connection to the VM. Note: If enabled for theroot user, key-based authentication can be used with the Linux/Unix scp command or tools like WinSCP on Windows to authenticate this account. I’m going to repeat the same general warning that I gave you in the secti on on enabling SSH via therootaccount: it’s a nice-to-have feature, fast, convenient, and all of that, but best practice in the real-world is to transfer files via a nonroot user if possible, and to elevate privileges to the root user only when really necessary. That being said, this is our lab, we’ve taken fairly strong precautions against shooting ourselves in the foot, we’re not running anything mission-critical, and we’re not storing any sensitive data. So, if you want to SCP stuff as root, have fun, know the risks involved, and be careful. 18 Network Design Factors When Working with baremetal Hypervisors So far, this book has provided a great deal of instruction on how to create virtual machines and how to set up virtual network infrastructures, yet almost all of the virtualization platforms covered in this document are hosted hypervisors, the only exception being VMware ESXi (VMware vSphere Hypervisor). Consider the following illustration: |||||||||||||||||||| |||||||||||||||||||| In the illustration above, our management workstation (a physical asset) is running a hosted hypervisor. This setup allows us to create a virtual network card on our host system, directly connected to the Management virtual network. This allows us to access virtual machines on the Management network easily, since the virtual network card and the virtual machines are considered to be on the same local network. The only virtual machine in the diagram above we need firewall rules and static routes to communicate with is the Kali Linux VM, which is on the IPS network. Compare that network diagram to this one: |||||||||||||||||||| |||||||||||||||||||| This illustration reflects a lab environment composed of a management workstation and a system running a bare-metal hypervisor, hosting the lab network, and all of the virtual machines (the cloud labeled VM Network) we want to access. As you can see, the management workstation, a separate physical system, has to be configured to route its traffic through the pfSense VM, which then has to be configured to allow access to the lab networks. In order to achieve this, we need to setup static routes for both of the lab networks, as well as configure firewall rules that allow our management workstation to access each of the lab’s VMs over the network, for the specific services we require access to. The upcoming sections detail what static routes and/or firewall rules management workstations require in order to interact with virtual machines in your lab when they are hosted on a baremetal hypervisor.. There is also a small section with regards to DHCP and how to configure static DHCP allocations on the physical network your management workstation, bare-metal hypervisor and pfSense WAN interface are deployed on to ensure that their IP addresses don’t change. Finally, there is a detailed section on how to deal with specific scenarios where your management workstation may be in a network where you cannot statically assign its IP address, or guarantee a static DHCP allocation. This section will define how to create either a physical or a virtual machine bastion host (referred to as a jump box). If required, this jump box, when configured with a static IP address or DHCP allocation, can be used to |||||||||||||||||||| |||||||||||||||||||| access the hypervisor, pfSense webConfigurator, and all of the VMs on the hypervisor through the use of SSH tunnels (which we will also be discussing in this chapter). 18.1 Prereqs This guide assumes that you are running ESXi 6.x or higher, with the latest updates available installed, with the ESXi lab environment configured as instructed in the VMWare vSphere Hypervisor setup guide (meaning all the virtual networks, as well as all the virtual machines presented in said section have already been created, and the pfSense VM can be accessed and managed via its webConfigurator UI). It is presumed that the Management and IPS virtual networks are set to 172.16.1.0/24 and 172.16.2.0/24, respectively. Most of the commands and illustrations in this guide assume that your management workstation is running Microsoft Windows 7, 8, or 10. However, for the OS X, Linux and BSD users out there, OS-specific commands and instructions will be provided on an as-needed basis. This guide also assumes that the management interface of the ESXi server, the WAN interface of the pfSense firewall VM, and the network card of the workstation (or Bastion Host/jump box) being used to manage your bare-metal hypervisor and the virtual machines hosted on it, all either have static IP addresses or static DHCP reservations. This is in order to ensure that your lab network and bare-metal hypervisor can be easily accessed and managed at all times. Just to reiterate, at an absolute minimum, you need to set up a static IP address or static DHCP allocation for your ESXi server’s management interface, the WAN interface of your pfSense VM, and for the network card of an additional system (whether that system is a management workstation OR a jump box) on a physical network that can access the ESXi server and the pfSense WAN interface in order to manage your lab network. 18.2 Creating Static Routes |||||||||||||||||||| |||||||||||||||||||| As mentioned above, managing a lab environment that is based on a bare-metal hypervisor, using a dedicated management workstation, requires the creation of custom network routes to be able to interact with the lab virtual machines over the network. On the network I used to create the labs for this guide, I assigned IP addresses from the 192.168.1.0/24 range to my physical machines. The WAN interface of the pfSense VM was given a static DHCP allocation, the IP address being 192.168.1.22. With this information, we can create static routes to the 172.16.1.0/24 and the 172.16.2.0/24 networks as necessary. If you’re using Windows 7 or greater, open an elevated command prompt, then execute the following commands, one at a time: route -p add 172.16.1.0 mask 255.255.255.0 192.168.1.22 route -p add 172.16.2.0 mask 255.255.255.0 192.168.1.22 Replace the IP address 192.168.1.22 in the statements above with one you assigned to the WAN interface of the pfSense firewall VM. To check the updated routing information of your system, just run: route print If you receive an output with entries similar to the ones in the illustration above, you have successfully added the static routes needed. Additionally, these routes should be persistant, meaning they will automatically be recreated on reboot. In case that you’re running OS X or a Linux distro on your management workstation, check out the corresponding Remote Access guide. Specifically, if you’re using Linux, pay attention to the sections detailing theip command, while OS X and BSD users can read up on theroute command. After having configured the necessary routes accordingly, you need to apply a method to ensure the routes persist upon reboot. Linux and BSD users should check out the section on using /etc/rc.local. Those of you working with OS X are referred to the section OS X Route Persistence for Bare-metal Hypervisors. |||||||||||||||||||| |||||||||||||||||||| 18.3 Creating Firewall Rules Note: If you plan on configuring a bastion host/jump box, you may want to skip over this section for now. Ensure that your management workstation is able to access the pfSense webConfigurator via the WAN IP address, and continue to the next section. In addition to the data packets being routed to the correct network, we have to ensure that the respective network traffic is actually permitted to reach its destination. To achieve this, we have to add firewall rules to the WAN interface to allow the traffic inbound. You need to log in to the web interface of your pfSense firewall VM via the WAN interface. If you followed the ESXi setup guide, you should have already created an allow firewall rule, via the easyrule utility, to permit your management workstation access to the webConfigurator UI. Didn’t do that? Check out the Network Configuration section for the pfSense VM to find out how to use pfSense’spfctl and easyrule command line utilities, in order to allow your management workstation to access the firewall from the WAN interface, or to add another management workstation, etc. Once you have logged in to the webConfigurator of the pfSense VM, navigate to Firewall > Rules, and ensure that the WAN interface is selected. All of the firewall rules we’ll be adding for remote access refer to the WAN interface. In our case, we have to add three firewall rules, each of them handling traffic for port 22/tcp (SSH). The destination IP addresses (use the single host or alias option) are 172.16.1.3, 172.16.1.4, and 172.16.2.2,for the SIEM IPS, and Kali virtual machines, respectively. The source IP address for each rule should be the one assigned to your management workstation, which in my case is 192.168.1.17 (again, use the single host or alias option, and, of course, be sure to modify the source IP address to match the one you did set on/for your management system). When you are done, your firewall rule set on the WAN interface should look similar to the one displayed in the picture below: The new rules should be placed BELOW the pass rule that allows your management workstation to access to the pfSense’s web interface, but ABOVE the explicit deny any/any rule at the bottom of the rule list (that is, in case you elected to create this rule). If you need to change the order of the rules to adjust the sequence of their evaluation, drag and drop the row(s) in question to their new position(s) in the list, then click the Save button. |||||||||||||||||||| |||||||||||||||||||| The easiest way to verify if those new firewall rules are working is to attempt to establish SSH connections to the SIEM, IPS and Kali virtual machines. Refer to the Remote Lab Management chapter to learn how to setup SSH connectivity to your virtual machines. Depending on the OS of your management workstation, follow the Windows Remote Access guide or the Linux, BSD, and OS X Remote Access guide, up to and including the respective section on configuring keybased authentication for your remote systems. Additionally, if you haven’t already, you may also want to have a look at the How to Enable SSH on Kali Linux guide, in order to be able to establish SSH sessions to your Kali Linux VM. If everything is working as intended, you should be able to establish SSH connections to the SIEM, IPS, and Kali Linux virtual machines, with the output of the respective sessions looking similar to those shown in the pictures below: |||||||||||||||||||| |||||||||||||||||||| As you can see in the illustrations, I was able to connect and authenticate successfully to each of the remote systems. At this point, if you haven’t already configured key-based authentication, I suggest that you visit the respective part of the the remote access guide that corresponds with your management workstation to do so. I also recommend reading the guide on how to disable password authentication on your lab virtual machines as well. management workstation to the machines of our lab environment, using the pfSense system. It is sensible to take precautions for the management system, too. If you’re using Windows as OS on your management workstation, you may want to read the guide on Defense in Depth for Windows Hosted Hypervisors, specifically the section Using Windows Firewall to Limit Exposure of Windows Hypervisor. Out of an abundance of caution, you probably want to make sure that you block ANY and all connections initiated from hosts on the 172.16.1.0/24, and 172.16.2.0/24 networks. 18.4 Dealing with DHCP Using DHCP to assign IP addresses to certain components of a lab environment can lead to remote access problems when dealing with VMs hosted on a bare- |||||||||||||||||||| |||||||||||||||||||| metal hypervisor. Setting up static routes and firewall rules requires the IP addresses that you’re routing to, as well as the source/destination addresses for the firewall rules to remain constant. As a part of the network setup portion for all of the hypervisors in this guide, we configured static DHCP mappings for all the virtual machines in our lab networks. This was to make sure that the IP addresses of each of our VMs will always remain the same, even after reboot, without having to set IP addresses, default gateways, or DNS information manually on each virtual machine individually. As mentioned above, if at all possible, your management workstation, ESXi server, and WAN interface of the pfSense firewall VM should either have static DHCP mappings based on the MAC addresses of these systems, or static IP addresses configured on the corresponding network interfaces themselves. Without knowing what your home or office network looks like, I can say with some measure of confidence that most network devices (even SOHO routers provided by most ISPs) allow you to configure static DHCP mappings or IP Address Reservations. The screen capture above was taken from my home network router, an Actiontec device my ISP provided me. Notice the option to set the DHCP lease type to static. If this thing is capable of static DHCP allocations, I’m sure that whatever gear you are using supports this functionality as well! Consult with the device documentation or your ISP’s tech support department if you need help. As an alternative, you can check the DHCP settings of your gear to identify the range of IP addresses used for the network(s) your workstation, ESXi Server |||||||||||||||||||| |||||||||||||||||||| and/or pfSense VM are connected to, and simply configure a static settings on the respective system’s NIC, using an IP address from the same local network, but outside of the DHCP address pool. For example, my home network has a SOHO router my ISP provides, with the internal network set to 192.168.1.0/24. The DHCP pool of the device is 192.168.1.2 - 192.168.1.100. This means that for my workstation, the ESXi server, and the WAN interface on my pfSense VM, I could use any IP address from the range 192.168.1.101 192.168.1.254 as static setting for each of the systems mentioned above. Bear in mind that if you set static IP addresses, you need to manually set the default gateway for these systems, as well as the DNS servers to use, too. 18.5 Jump Boxing Note: The remainder of this chapter details how to create and configure a jump box for accessing your lab environment, using a Raspberry Pi, an additional virtual machine, or any other physical hardware you happen to have available. If you were able to configure your management workstation with a static DHCP address allocation to ensure access to your lab, you may opt to skip over the sections related to jump boxing entirely. However, if your lab is intended to support incoming connections from multiple users and/or machines, knowing how to set up a jump box would be a more scalable solution for allowing multiple users to access to the virtual machine lab more reliably. What are you supposed to do when your management workstation resides in a network where static network interface configuration or static DHCP address allocations are not available? |||||||||||||||||||| |||||||||||||||||||| In most enterprise networks, end-user systems usually reside in a subnet and/or vlan that is configured to use DHCP. Unless you are VERY good friends with your network admins, you probably won’t be able to get a static DHCP mapping for your workstation. There may be technical reasons, or it may just be a policy. However, on the server subnet, the network and system admins usually have static DHCP address allocations or fixed IP address settings configured for the attached network interfaces. With that as a starting point, we can install a small embedded system (e.g. a Raspberry Pi), or create another small VM, which then can be connected to the server subnet, just like the ESXi server’s management interface, and the pfSense VM’s WAN interface. By adding firewall rules to let this system access the virtual lab networks, and by setting up the necessary static routes on this machine, we are able to redirect the traffic from our management workstation through this system into our lab environment. A system used to forward and broker connections from one network to another network in this manner is known as a bastion host, or a jump box. Note: Jump boxes are also referred to as proxies or redirectors. These names are interchangeable. Jump boxes are used as a connection broker to forward your network connections to another destination you cannot or do not want to interact with directly. Jump boxes are used by IT and information security staff to access assets in other network segments securely, in a manner that allows connections to/from sensitive networks to be audited (if needed), with the jump |||||||||||||||||||| |||||||||||||||||||| box acting as both a choke point, and a way to audit connections to sensitive assets. In the diagram above, the large, arrow-headed solid lines denote a tunnel connection from our management workstation to the jump box, which is located on the same subnet/vlan as our ESXi server. The lighter colored lines to the ESXi server (more precisely to the pfSense VM on the ESXi server) and the VM lab network represent connections from our management workstation that were forwarded from the jump box system to destinations in the VM lab. 18.6 Using a Raspberry Pi as a Jump Box I had a Raspberry Pi Model B (revision 1) from some time ago at hand, so I wrote this guide using this hardware, specifically. The process described in the following should be nearly identical for most of the Raspberry Pi models, especially for those that come equipped with an Ethernet port. This guide also assumes that you’ll be using wired Ethernet to connect to your Raspberry Pi jump box, so if you want to use WiFi, you’ll be on your own (though there shouldn’t be too much of a difference, really). Why go through the trouble of setting up a physical jump box, you ask? Having a physical jump box has some advantages to it that you should consider. A lot of newer servers and server motherboards come with small, embedded |||||||||||||||||||| |||||||||||||||||||| systems integrated into their chassis, often referred to as LOM, IPMI or some variation. As long as the server itself is plugged in and getting power, those integrated systems can be accessed over the network, providing a variety of functions that normally require physical access to a machine. Among other things, they make it possible to power the server on or off physically, and to redirect the server’s console output over the network. Having a separate physical jump box not only allows us to access our lab’s VM systems, but to connect to and use these management systems, too. For example, if the power on the server goes out, and you can’t physically access this machine, having a physical jump box at least gives to possibility to try to troubleshoot the issue remotely (e.g. virtually pushing the power button etc.). While you don’t have to use a Raspberry Pi specifically, one of its advantages that deserves attention is that it doesn’t require much in the way of power, which is supplied via a micro USB connector. Using a USB to micro USB cable, you can choose to power the Raspberry Pi by connecting it to the server itself, or by hooking it up to an AC adapter. In either case, the Raspberry Pi requires very little power to operate, making it an ideal physical jump box. 18.7 Installing the Raspbian Image to your Raspberry Pi The fine folks at raspberrypi.org have made available a great guide for doing an initial setup of the Raspberry Pi. We’re going to follow their instructions for installing Raspbian, an RPi (shorthand for Raspberry Pi) optimized version of the Debian Linux distro, onto an SD card, which we will use to boot the tiny computer. You can use any SD card you’d like, so long as it is 4GB or larger. In my case, I used a spare 8GB SD card I found lying around. The first thing recommended in the guide is reformatting the SD card. The instructions for doing this are located at . I was using a machine running Windows,, so I downloaded the suggested SD card formatter tool from sdcard.org/downloads, then reformatted my card. (Please note that the page link above provides instructions for formatting your SD card on both OS X and Linux too, suggesting the use of the SD card formatter tool for OS X, and of thegparted utility for Linux, respectively.) Be sure to configure the Format size adjustment option to ON in order to resize the SD card partition to make use of the entire capacity of the card if necessary. |||||||||||||||||||| |||||||||||||||||||| Note: If you are using an SDXC card (these are defined as SD cards over 32GB in size), make sure to follow the instructions on how to format this kind of cards on Windows, Linux, and OS X, available at . Next, visit to download the latest version of Raspbian (Note: Raspbian Jessie was the version used in this guide). I recommend downloading the lite image, because apart from the initial setup that we are going to do together, you’ll probably never connect a keyboard and monitor to your RPi again (unless something breaks), let alone using the jump box as a regular Linux desktop machine. Anything that can be done to save drive space, RAM and CPU cycles is worthwhile, which makes the lite image the perfect choice. |||||||||||||||||||| |||||||||||||||||||| After downloading the file, you need to write the image to the SD card. There are a number of image writing utilities for a variety of operating systems. The official instructions on raspberrypi.org suggest to using a graphical tool called Etcher, available from the website etcher.io that is available for Windows, Linux, and OS X, which from my personal experience fits the bill nicely. Simply select the image file, select the destination disk (the SD card), and click the Flash! button to write the image to the drive and have it verified after it has been written. Note: in the past, you would normally have to unzip the downloaded image file prior to writing it to an SD card. etcher is capable of unzipping the compressed image file automatically, making this process no longer necessary. Now comes the fun part. Install the SD card into your Raspberry Pi, hook up a USB keyboard, Ethernet cable (make sure that it is attached to the same network that the ESXi server and pfSense VM’s WAN interface are available on), and a HDMI video device to the mini computer, and finally connect USB power to start it up. |||||||||||||||||||| |||||||||||||||||||| |||||||||||||||||||| |||||||||||||||||||| |||||||||||||||||||| |||||||||||||||||||| Booting the system takes a few moments. The output on your screen should look similar to the one shown in the picture above. If you are finally greeted with a login prompt, given you have everything connected to it, your Raspberry Pi is ready to go. 18.8 Configuring Raspbian The default credentials to log in to Raspbian are username pi and password raspberry. Enter these credentials to log in via the console, using the keyboard and monitor you attached to the Raspberry Pi. In order to use it as our jump box, we need to make sure that the RPi system has an IP address, verify the its MAC address for creating a static DHCP mapping, and enable Raspbian’s OpenSSH service. Once you have logged in, run the command ifconfig -a to display information about the network interfaces on the RPi. GIven that there is only one physical network adapter present, look for the block of information that starts with eth0, note the corresponding MAC address (labeled HWaddr in theifconfig output), and the currently assigned IP address (namedinet addr). In the picture above, taken from the screen of my system, you can see that the |||||||||||||||||||| |||||||||||||||||||| RPi has an IP address of 192.168.1.10, and a MAC address of b8:27:eb:b3:77:a6. Document the MAC address displayed in order to configure a static DHCP mapping for this system. I would advise setting up a static DHCP mapping now, or as an alternative, configure a static IP address for the interfaceeth0 on this system. Revisit the section Dealing with DHCP on instructions on how to setup static DHCP address allocations. Next, run the commandsudo raspi-config. A menu captioned Raspberry Pi Software Configuration Tool (raspi-config) will pop up, offering a variety of different choices. Select option 7 Advanced Options, followed by option A4 SSH on the next page to enable the SSH service. After enabling the SSH server, you are returned to the main menu. Hit ESC or select Finish to exit the Raspberry Pi Software Configuration Tool and gain access to the command line again. Typeexit to logout of the Raspberry Pi’s |||||||||||||||||||| |||||||||||||||||||| console.Next, we have to test if the SSH server is available now and confirm that it is possible to log in remotely to the RPi. Using the IP address you recorded from theifconfig output, try to establish an SSH connection to the Raspberry Pi, using your SSH client of choice (e.g. mRemoteNG if you are using Windows, or thessh command in case you are on Linux, BSD, or OS X). Verify that you can log in using the system’s default username and password. The picture below shows the mRemoteNG Connections item I used for my test. Linux/OSX/BSD users should run the commandssh [Raspberry Pi’s IP address] to verify SSH connectivity. Note: If you are able to login successfully via SSH, the system will inform you that you have NOT changed the default password for the pi user yet. We will be fixing this momentarily. Now that you have remote access to your Raspberry Pi, there are some things that need to be taken care of before moving on. To do so, open the Raspberry Pi Software Configuration Tool again by running the |||||||||||||||||||| |||||||||||||||||||| commandsudo raspi-config. In the main menu, as depicted below, select option 1 Expand Filesystem. This will expand the Raspbian image to utilize the full capacity of the SD card in order help prevent the system from running out of disk space. On selecting this option, the system informs you that the partition will be re-sized on the next reboot. Continue by selecting option 2 (Change User Password) from the main menu, and change the pi user’s password to a more secure one. Even though we are going to password authentication via SSH, it is good security hygiene to never leave default credentials enabled on any system you use. When you are done, back in the Raspberry Pi Software Configuration Tool’s main men u, hit ESC or select Finish to exit. The system will ask if you want to reboot now. Select Yes, then give the RPi some time to go through the reboot. Afterwards, reconnect to your Raspberry Pi via SSH. Remember to use the new password you assigned to thepi user. Note: It is very important that you set a complex password for thepi user. The pi user has the ability to use sudo and run commands as root without a password. Take extra care to secure this account’s password, and ensure that |||||||||||||||||||| |||||||||||||||||||| you run through the section on disabling password authentication for SSH. Once you have logged back into the RPi over SSH, you’ll want to update the system and its packages by running the command line export DEBIAN_FRONTEND=noninteractive; aptget update; apt-get -y dist-upgrade. I also highly recommend utilizing the updater.sh script from the Automated Patching for Linux Lab VMs section for ensuring that your RPi jump box continues to stay up to date with the latest patches and security updates. At this point, you can safely remove the monitor and keyboard from the RPi, thus making your jump box headless. After you have finished updating your RPi system, review the Setting Up Your Jump Box section for the next steps. 18.9 Creating a Jump Box VM As I explained in the section on using a Raspberry Pi as a jump box, using dedicated physical hardware holds some advantages. However, in the event that you simply do not have a Raspberry Pi (or other, physical hardware) available, running a VM to serve as your jump box is also an option. Start by creating a small Ubuntu Linux 64-bit VM on your ESXi server. Assign it 512mb of RAM, 10GB of hard drive space (thick provisioned, attached to SATA Controller 0 (0:0)), and a single virtual network card connected to the Bridged network. The picture below shows the suggested virtual hardware setup: |||||||||||||||||||| |||||||||||||||||||| Just like for the majority of our virtual lab’s Linux systems, install Ubuntu Server 16.04. Use t he ISO you previously uploaded to the datastore. While the installer is running, check the hardware configuration of your VM from the ESXi web interface, and record the MAC address of your VM, if you plan on configuring a static DHCP address allocation for your jump box. |||||||||||||||||||| |||||||||||||||||||| Follow the same OS installation instructions as you did for the other Ubuntu systems on your ESXi host. Make sure to configure the host to receive security updates automatically, and that the OpenSSH server, and the standard system utilities packages, are installed. After the installation is complete, power down the VM, and edit the hardware settings. Make sure to remove the USB controller, CD/DVD drive, and the SCSI controller. |||||||||||||||||||| |||||||||||||||||||| Power the VM back on, login to your new VM, and download the latest updates for it by executing the command sudo su to become root, followed by running export DEBIAN_FRONTEND=noninteractive;apt-get -q update;aptget -y -q dist-upgrade. As soon as your system is up to date, run ifconfig -a to display the IP address (labeledinet addr) that was assigned to your system, as well as to confirm your system’s MAC address (taggedHWaddr). In the image below, showing the console output of my system, you can see that the VM has an IP address of 192.168.1.8, and a MAC address of 00:0c:29:c1:ea:4a. If you did not record the MAC address of your system earlier, and used it to create a static DHCP mapping for this VM’s network interface on your physical network, I would advise doing so now. Revisit the section Dealing with DHCP on instructions on how to setup static DHCP address allocations, or as an alternative, configure a static IP address for this interface. |||||||||||||||||||| |||||||||||||||||||| Afterwards, confirm that you are able to establish an SSH connection to the VM using your SSH client of choice (e.g. Windows mRemoteNG if you are using Windows, or the ssh command in case you are on Linux, BSD, or OS X). Verify that you can log in with the username and password you created during the OS installation. While you’re here, you may also want to implement the updater.sh script mentioned in the Automated Patching for Linux Lab VMs section to ensure your jump box is always up to date with the latest patches. After you are finished, complete the Preparing Your Jump Box for Service section. 18.10 Other Physical Jump Boxes If you prefer a physical machine over a VM as a jump box but don’t have a |||||||||||||||||||| |||||||||||||||||||| Raspberry Pi available, you can easily substitute the latter with any spare hardware you have at hand. Follow along with instructions from the Creating a Jump Box VM section (ignoring the ESXi configuration portions), and simply install Ubuntu Server, with the prescribed installation packages, on your physical jump box. The major differences to note are that you need to physically plug in your jump box to the same physical network the ESXi server and pfSense VM’s WAN interface reside on, and that you will need to wait until the operating system is installed to check the MAC address of your system for making a static DHCP address allocation. Other than that, the process is more or less identical. 18.11 Preparing Your Jump Box for Service This section instructs you on how to finish setting up your jump box, and how to configure it as well as the management workstation to enable access to your VM lab. This portion of the guide borrows very heavily from other material in this book, namely a vast majority of the Remote Lab Management guide. For example, setting up SSH on Windows or Linux/OS X/BSD management workstations (including recommended applications, generating SSH public/private key pairs, copying the public keys to other systems for keybased authentication, and finally, disabling password authentication as a safety measure to prevent unauthorized access), and setting up persistent static routes on the jump box. The following passages also cover setting up static DHCP address allocations, creating firewall rules to allow your jump box to access the lab VMs, and finally, how to the jump box to forward network traffic to the lab VMs via SSH Tunneling (also known as TCP Forwarding). 18.12 Configuring Static DHCP Address Allocations I highly recommend setting up your jump box to utilize a static DHCP address allocation, if at all possible, as opposed to manual configuration. Which options you have available for implementing static DHCP address allocations on your network depends on where you set up your lab (e.g. home vs. work), and the network equipment you’re using, allowed to use and able/allowed to access/configure. In my case, the router on my home network allows me to configure static |||||||||||||||||||| |||||||||||||||||||| DHCP address allocations via an option called IP Address Distribution. Among other things, this option lets me display current DHCP leases, and change the type of lease from dynamic to static allocations easily, as you can see in the image below. I was able to confirm that my host, named “jumpbox”, with the MAC address I recorded earlier was assigned the IP address 192.168.1.8. All I had to do to turn this entry into a static DHCP address allocation was check the Static Lease Type checkbox and click Apply. Depending on the system serving DHCP on your home or office network, configuring static DHCP allocations may be just as simple, or more complicated. 18.13 Enabling Key-Based Authentication for your Jump Box The next step is to configure this system for remote access over SSH, specifically, copying your public SSH key to the host, and disabling password authentication on the jump box via /etc/ssh/sshd_config. I will go over doing this from both Windows and Linux/OSX/BSD hosts. 18.14 Windows If you intend to connect to your jump box using a management workstation that is running Windows, follow the Windows Remote Access guide, skipping over the section on persistent static routes. Specifically, take care of the following: 1. You should have already used mRemoteNG to log in via the user you set up during the OS installation for your jump box. If you have not, download and install mRemoteNG, create a Connections item (see picture below for reference), and establish an SSH connection to the jump box. |||||||||||||||||||| |||||||||||||||||||| 2. You should already have an SSH public/private key pair generated via PuTTYgen. If that is not the case, follow the corresponding section of the Windows Remote Access guide to do so now.. Add the SSH public key to the ~/.ssh/authorized_keys file on the jump box. 3. Modify your PuTTY Session and mRemoteNG connection profile to log in to the jump box with your SSH key, and verify that key-based authentication is working (make sure to remove the password from your mRemoteNG connection profile). If you need assistance with this, the Windows Remote Access guide has detailed instructions for creating a new PuTTY session that utilizes your SSH key. |||||||||||||||||||| |||||||||||||||||||| 4. I highly recommend that you disable password authentication for SSH connections on your jump box entirely due to its exposure to the physical network. If you need guidance on how to do this, refer to the Disabling Password Authentication Entirely via sshd_config section of the Remote Lab Management guide. Become the root user (sudo su) and modify/etc/ssh/sshd_config to disable password authentication, then restart the SSH service. At this point, you should be able to connect to the jump box from your Windows workstation via SSH with no password required. 18.15 Linux/OS X/BSD If you plan to connect to your jump box via Linux, OS X, or BSD as your management workstation you will use to connect to your jump box, follow the Remote Access for Linux and OS X guide, skipping over the sections on persistent static routes. Specifically, |||||||||||||||||||| |||||||||||||||||||| take care of the following: 1. You should have already used ssh to log in via the user you set up during the OS installation for your jump box. If you haven’t already, I suggest creating an alias for establishing SSH connections to your jump box (e.g.echo “alias jumpbox=’ssh ’” >> ~/.bash_profile; source ~/.bash_profile). Try to connect to the system with the new alias to make sure it works. 2. Add your SSH public key to the remote user’s ~/.ssh/authorized_keys file on the jump box. 3. Close the current connection, then try to reconnect to the jump box using the alias you made before to see if key-based authentication is working properly. |||||||||||||||||||| |||||||||||||||||||| 4. I highly recommend that you disable password authentication for SSH connections on your jump box entirely due to its exposure to the physical network. If you need guidance on how to do this, refer to the Disabling Password Authentication Entirely via sshd_config section of the Remote Lab Management guide. Become the root user (sudo su) and modify/etc/ssh/sshd_config to disable password authentication, then restart the SSH service. At this point, you should be able to connect to the jump box from your Linux/OS X/BSD workstation via SSH with no password required. 18.16 Adding Static Routes to your Jump Box After verifying that we can connect to it from our management workstation, we need to set up the jump box to access the VM lab network. This section will be borrowing material from Remote Access for Linux and OS X, since we’ll be configuring a Linux system (our jump box system) to access our VM lab networks. Specifically, we’ll be using the passages on theip command and/etc/rc.local to create persistent static routes on reboot. |||||||||||||||||||| |||||||||||||||||||| Connect to your jump box over SSH, then enter the sudo su command to become the root user. Using vi or the command line text editor of your choice, you need to modify the file /etc/rc.local, in particular to add the commands necessary to establish static routes to the 172.16.1.0/24 and 172.16.2.0/24 networks. In the screen capture above, 192.168.1.22 is the IP address of the WAN interface of the pfSense virtual machine on my lab network. Adjust that IP address accordingly for your network. After you have finished editing the /etc/rc.local file as the root user, reboot your jump box, then reconnect to it over SSH. Next, run the command ip route show to display the contents of the routing table. |||||||||||||||||||| |||||||||||||||||||| As you can see in the image above, rc.local has successfully added static routes to the VM lab networks on the jump box system. The output you received should look similar, with only the portion to the right ofvia being different. 18.17 Adding Firewall Rules and SSH tunnels to allow access to the VM lab networks Within the VMware vSphere Hypervisor (ESXi) Setup guide, located in the portion on network configuration for pfSense, there is an initial walk-through on how to setup and maintain access to the pfSense web interface from the WAN interface of the firewall. The respective procedures invoke thepfctl andeasyrule commands, issued from the console of the pfSense VM. We use theeasyrule command to create a permanent firewall rule to allow our management workstation persistent access to the pfsense webConfigurator UI via the WAN interface. Adding this rule alone is not enough, however. By default, pfsense blocks RFC1918 addresses on the WAN interface. This is because usually, firewalls are deployed at the perimeter of a home, or enterprise network and the WAN interface is assumed to be connected to a public, routable network (e.g. the internet). RFC1918 address are not publicly routable, and therefore in that case, the default action of dropping traffic from RFC1918 addresses on the |||||||||||||||||||| |||||||||||||||||||| WAN interface is reasonable. In our case however, the WAN interface of our pfSense firewall is connected to an RFC1918 network, as is our management workstation we will be using to manage and configure the firewall (and all of our lab virtual machines). Because this default RFC1918 rule takes precedence over any rules we add via easyrule or the web interface, we must also use pfctl -d to temporarily disable the firewall in order to login to the webConfigurator and permanently disable the default setting of blocking RFC1918 IP addresses on the WAN interface, to let your management workstation connect to the webConfigurator UI. In addition to the firewall rule above, in the pfSense Firewall Rules and Network Services Guide, there is a section dedicated to Creating Firewall Rules to enable SSH access to your lab VMs from your management workstation (Specifically, Firewall Rule Configuration - Bare-metal Hypervisors). We will simply be applying these firewall rules to our jumpbox system, as opposed to our management workstation. If you are still able to access the the pfSense webConfigurator UI from your management workstation, you may simply go ahead creating new firewall rules for your jump box system in order to allow it to access the VM lab. Otherwise, if you cannot or are no longer able to reach the web UI, you have to usepfctl -d from the console of the pfSense virtual machine first, to temporarily disable the firewall, to enable yourself to set up the rules necessary via the webUI. 18.18 I Can Still Access the pfSense webConfigurator with my Management Workstation Log in to the pfSense webConfigurator, then navigate to Firewall > Rules, and if it isn’t already highlighted, select the WAN interface. |||||||||||||||||||| |||||||||||||||||||| As you can see from the screen capture above, I have a number of rules that allow my Windows workstation (192.168.1.17) access to the pfSense webConfigurator over HTTPS, and several of the lab VMs over port 22 (SSH). We’re going to add a new set of rules to allow the jump box system (which in my case is configured with the IP address 192.168.1.8) as well to access those systems. Modify the firewall policy for the WAN interface by adding 4 new pass rules: one to allow access the pfSense WAN IP address (e.g. 192.168.1.22 in my case) over port 443/tcp(https), and 3 rules to enable access to the Kali VM (172.16.2.2), the SIEM VM (172.16.1.3), and the IPS VM (172.16.1.4) over port 22/tcp(SSH) respectively. These new rules must be placed ABOVE the explicit deny any/any entry (that is, if you chose to create one. By default pfSense denies any traffic on an interface that is not explicitly allowed). When finished, your list of firewall rules for the WAN interface should look similar to the one shown in the picture below, with the values of the source IP addresses probably differing from the one depicted: |||||||||||||||||||| |||||||||||||||||||| At this point you can log out of the webConfigurator from yo ur management workstation. Don’t delete the existing firewall rules for your management workstation yet, and don’t take a new snapshot of the pfSense VM at this point, because we need to verify that the extended firewall policy in general, and the new rules in particular, are working as intended. To find out if they do, go to the section TCP Forwarding and You, then Testing your Dynamic Tunnels with FoxyProxy. After having verified that your dynamic tunnel is working and having logged on from the IP address of your jump box, proceed to Testing Your Forward Tunnels. After having testing connectivity to all of your lab assets using your jump box system, return to the web UI of your pfSense VM. On the page that lists the firewall rules for the WAN interface, use the icon that looks like a trash bin to delete the rows referring the IP address of your management workstation (e.g. in the above illustration, that would apply to all the rules containing the IP address 192.168.1.17). After cleaning up, your final ruleset should look similar to the one displayed in the image below: |||||||||||||||||||| |||||||||||||||||||| Once you have finished modifying the firewall rules for the WAN interface, save your changes and log out of the webConfigurator. If everything has gone well so far, your jump box should be ready for service. 18.19 I Have Lost Access to the pfSense webConfigurator UI So you no longer have access to the webConfigurator interface, either due to misconfiguration, or due to DHCP giving your workstation a different IP address. As long as you are still able to access the ESXi server via its web interface or by utilizing the VMware vSphere Client from your management workstation, that is not a problem. Use one of the two options mentioned above to connect to the ESXi host. (For the sake of simplicity, the following instructions are based on the web UI.) Click on Virtual Machines in the Navigator panel on the left side of the browser window, right click on the pfSense VM. From the context menu, choose Console, then click Open browser console. A console session to the pfSense VM appears in your web browser, just like the one we used to initially configure this system. |||||||||||||||||||| |||||||||||||||||||| Click inside the console session, then select option 8 to access the shell interface of pfSense. Enter the commandpfctl -d to completely disable the firewall. This is the fastest and easiest way for us to regain access to the system. Don’t worry, we’ll re-enable it in just a moment. At this point, the firewall is disabled. We need to establish a dynamic tunnel and proxy off of our jump box to continue. Please proceed to the section TCP Forwarding and You, then Testing your Dynamic Tunnels with FoxyProxy. After working through those passages of the guide, return here to complete the configuration of the firewall. After successfully setting up and testing the TCP forwarding/SSH tunnelling, confirm that you can login to the pfSense webConfigurator from your management workstation via the WAN interface of the pfSense VM. Navigate to Firewall > Rules, and if it isn’t already highlighted, select the WAN interface. We’re going to add a new set of rules to allow the jump box system (which in my case is configured with the IP address 192.168.1.8) to access the pfSense webConfigurator over HTTPS (443/tcp), and several of the lab VMs over port 22/tcp (SSH). Modify the firewall rules for the WAN interface by adding 4 new pass rules: |||||||||||||||||||| |||||||||||||||||||| one to access the pfSense WAN IP address (e.g. 192.168.1.22 in my case) over port 443/tcp(https), and 3 rules to enable access to the Kali VM (172.16.2.2), the SIEM VM (172.16.1.3), and the IPS VM (172.16.1.4) over port 22/tcp(SSH) respectively. These new rules must be placed ABOVE the explicit deny any/any entry (that is, if you chose to create one, by default pfSense denies any traffic on an interface that is not explicitly allowed). Depending on whether or not you had added firewall rules to allow the (presumably now invalid) IP address of your management workstation access to your lab network, and apart from the actual values of the source IP addresses, your list of rules for the WAN interface should look similar to the one displayed in the image below: Using the icon that looks like a trash bin, delete the rows referring the (old) IP address of your management workstation (e.g. in the above illustration, all the rules containing the IP address 192.168.1.17). After cleaning up, your final ruleset should look similar to the one shown in the picture below: |||||||||||||||||||| |||||||||||||||||||| Once you have finished modifying the firewall rules for the WAN interface, save your changes and log out of the webConfigurator. On saving the firewall configuration, pfSense should have re-enabled the firewall automatically. However, if you want to be 100% sure about that, log in to the ESXi server via its web interface and open a console session to the pfSense VM. Select option 8 to open a root shell on the system, then run the command pfctl -e. If you received an output similar to the image above, everything worked as expected and the firewall has already been re-enabled. If the status message on your system looks any different, you almost certainly didn’t save the recent firewall policy changes. In that case you probably need to runpfctl -d once more, and reconfigure the rules on the WAN interface again. If everything has gone well so far, proceed to the Testing Your Forward Tunnels section. As soon as the forwarding tunnels are working properly, your jump box should be ready for service. 18.20 TCP Forwarding and You Most SSH servers have an option called TCP Forwarding available. A lot of IT professionals also refer to this capability as SSH Tunneling. The idea is that SSH can be used to forward TCP network connections, not unlike a proxy, resulting in TCP connections to appearing as though they are being sent from the SSH server itself. This functionality is the core of what will enable us to |||||||||||||||||||| |||||||||||||||||||| use our jump box to access the systems inside our VM lab. You need to make some modifications to the SSH sessions you created to access your bastion host earlier. If you are using a management workstation running Windows, review the Windows SSH Tunnels section. If you are using Linux/OS X/BSD, check the Linux/BSD/OS X SSH Tunnels section instead. These guides will teach you how to create dynamic SSH tunnels for forwarding web traffic through your jump box, and how to forward SSH connections (as well as those based on other protocols) to the lab VMs as well. 18.21 Windows SSH Tunnels If you created mRemoteNG SSH sessions earlier to use key-based authentication, most of the following setup instructions will seem VERY familiar. Start up the mRemoteNG program, then on the menu bar at the top of the screen, select Tools > Options to open the corresponding window. Once in the Options menu, select Advanced on the left pane, then click on the Launch PuTTY button on the right side of the window. |||||||||||||||||||| |||||||||||||||||||| This should open up a window titled PuTTYNG Configuration. (The PuTTY executable is employed by mRemoteNG to establish SSH connections.) Under the Category pane, click on the plus sign next to SSH, then click to select the Tunnels entry. |||||||||||||||||||| |||||||||||||||||||| Enter a random number (above 1024) into the box next to the label Source port. This value sets the TCP port that the Windows workstation will bind to when a connection is created based on these configuration settings. I recommend picking a port number above 30000, like 31337, or another random, high-numbered TCP port. Leave the input box named Destination blank, click the radio button next to Dynamic, make sure that the one next to Auto is set, then finish by clicking the Add button. If you did it right, the value you entered for the source port, expanded with a leading “D”, should appear in the message box labeled Forwarded ports:. In my case, I chose 31337 for the source port, resulting in the value “D31337” being displayed in said box, as you can see in the picture below. |||||||||||||||||||| |||||||||||||||||||| While still in the PuTTYNG Configuration tool, we’re going to add three more tunnels that will allow us to access our Lab VMs more easily. Start by entering another random number in the Source port input box; for instance, I chose 31155. Next, enter “172.16.1.3:22” into the Destination input box. Make sure that the Local radio box is selected, then click the Add button. Repeat this process twice, using a different value for the source port each time (I used ports 31156 and 31157), setting the destination to “172.16.1.4:22” and “172.16.2.2:22”, respectively. Your Forwarded ports: field should look similar to the ones in the pictures below: Note: The entry D31337 should be there, you just might have to scroll down in the Forwarded ports field to reveal it. There should be FOUR entries in total. This configuration creates a new PuTTY session specifically for connecting to |||||||||||||||||||| |||||||||||||||||||| our jump box with one dynamic TCP tunnel, and three forward connect TCP tunnels. The dynamic tunnel allows us to use our jump box as an HTTP/HTTPS proxy, employing its network connection to interact with the pfSense webConfigurator, while the forward tunnels enables us to send SSH connection requests to our lab VMs via the SSH service of our jump box. Continue the configuration setup by selecting on the entry labeled Auth in the Category pane, then click the browse button next to the input box titled Private key file for authentication:. In the Explorer window that opens up, browse to the private key file (the one with a “.ppk” extension) you created with PuTTYgen earlier, and double-click select it. On return to the PuTTYNG Configuration window, the input box should populate with the path to your private key file. Finally, scroll up on the Category pane and select Session. Find the box titled Saved Sessions, enter a short description for this PuTTY connection profile (I entered “Bastion_Tunneling”), then click the Save button. Finish this setup by clicking the Close button to exit PuTTYNG Configuration. |||||||||||||||||||| |||||||||||||||||||| Back in mRemoteNG, close the Options menu window to return to the main screen. Under Connections, select the entry you created for your jump box. In my case, the session is named “jumpbox”. In the Config portion, under Protocol, find the entry called PuTTY Session. When clicked on, a small box with a downward facing arrow appears. Click on this arrow to get a dropdown menu, listing all the PuTTY sessions mRemoteNG is configured with. Click on the one you previously created (which in my case is Bastion_Tunneling) to

select it. |||||||||||||||||||| |||||||||||||||||||| Finally, click File > Save Connection File to save your changes. Double click on your newly modified mRemoteNG session for your jump box to see that it works and you are able to authenticate. If everything was configured correctly, you should be greeted by a screen similar to the one in the pictured below. If the connection attempt worked without problems, open up a command prompt on your Windows host, then run the following: netstat -ano | findstr 127.0.0.1:31 Note : You might have to change the trailing “31” if you entered different values for the source port setup in your recently created PuTTY session configuration. In that case replace the 31 with the first two digits of the numbers you used. |||||||||||||||||||| |||||||||||||||||||| The resulting output (which for my setup looked like the one in the image above) shows all lines returned by thenetstatcommand that contain “127.0.0.1:31”. Since all of the current connection’s tunnels open a port on 127.0.0.1 (localhost) on TCP ports 31155-57 and 31337, I searched for “31”, which is the common denominator of the TCP ports I defined in the Bastion_Tunneling PuTTY session on my machine. Since the command has returned four entries, it appears that the tunnels are all functioning normally. When you are finished here, you can proceed to Testing your Dynamic Tunnels with FoxyProxy. 18.22 Linux/BSD/OS X SSH Tunnels Earlier, on our management workstation, we created an alias to establish SSH connections to our jump box and saved it to our user’s~/.bashrc or~/.bash_profile configuration. Before you make any change to this file again, I highly recommend making a backup copy by running cp ~/.bash_profile ~/.bash_profile_bak; cp ~/.bashrc ~/.bashrc_bak just to cover your bases and make sure you have something to recover from in case your changes don’t work as expected. Next, using your favorite text editor, you need to modify the alias entry for your jumpbox system to: alias jumpbox=’ssh -D31337 -L31155:172.16.1.3:22 -L31156:172.16.1.4:22 L31157:172.16.2.2:22 ’ Note: In the alias above, the ports 31337, 31155-57 can be any TCP port above 1024, as long as the port is NOT currently in use by the operating system, and the port number chosen for each -L or-D option is NOT the same. I recommend using TCP ports 30000 or above. As a reminder, you may need to modify the finishing portion of yourssh alias to the username and IP address you configured for jump box in your environment. This alias tries to establish an SSH connection to the bastion host, including |||||||||||||||||||| |||||||||||||||||||| one dynamic (-D) TCP tunnel, and three Local (-L) forward connect TCP tunnels. The dynamic tunnel mentioned above allows us to use our bastion host as a proxy, employing its network connection to interact with the pfSense webConfigurator, while the forward tunnels enables us to send SSH connection requests to our lab VMs via the SSH service of our jump box. Save the changes to your .bash_profile or.bashrc file, then either use thesource command (e.g. source ~/.bashrc ORsource ~/.bash_profile) to reload the configuration, or simply reboot your management workstation. After that, verify you can connect to your jump box by using the adjusted alias. If the connection at tempt worked without problems and you’re using OS X/BSD, open a terminal session, then enter and execute the following: netstat -anp tcp | grep LISTEN | grep 127.0.0.1.31 Note: You might have to change the trailing “31” if you entered different values for the ports in your alias. In that case replace the 31 with the first two digits of the numbers you used. This command runs netstat -anp tcp, which in this case, is exclusively looking for current TCP connections. We then use the grep command to check the results returned by the previously run program for lines that containLISTEN |||||||||||||||||||| |||||||||||||||||||| (as in that port is listening for connections). We then usegrep again to look for any lines within the results from the earlier execution that contain “127.0.0.1.31”. This specifically looks for TCP sessions listening for connections on 127.0.0.1 (localhost). All of the TCP ports defined in the example ssh alias command above begin with “31”, resulting in thegrep command matching on 31155-31157 and 31337. Since the command has returned four entries, it appears that the tunnels are all functioning normally. Linux users should run: netstat -tlpn | grep 127.0.0.1:31 to receive similar results. Just like with OS X/BSD systems, you are expecting to have four lines returned, similar to what was explained above. When you are finished here, proceed to Testing your Dynamic Tunnels with FoxyProxy. 18.23 Testing your Dynamic Tunnels with FoxyProxy Before we move on to actually testing the dynamic tunnels we just prepared, we need to configure our management workstation’s web browser to use the jump box as a proxy. Since they both support the browser extension FoxyProxy, I recommend using either the current version of the Google Chrome or Mozilla Firefox web browser. The FoxyProxy extension allows you to create multiple proxy configurations/sessions that can be easily switched between, and instantly applied to your web browser at any time. With the help of this tool and the necessary settings in place, we will be able to use our management workstation to access the pfSense webConfigurator via the dynamic SSH tunnel established to our jump box. For this guide, I used Mozilla Firefox on Windows 10, however, aside from the initial installation of the add-on and how to access it, configuring FoxyProxy for utilizing our dynamic tunnel should be nearly identical for users of both Chrome and Firefox, on Windows as well as on OS X, Linux and BSD. Note: Use of FoxyProxy, Google Chrome, or Mozilla Firefox isn’t mandatory in order to use our Dynamic SSH tunnel as a web proxy, this just happens to be the easiest way to do so, and the method I recommend. Most modern web browsers have configuration options that allow users to manually specify a proxy to connect to. Pay attention to the detail that the dynamic SSH tunnel should be registered as a SOCKS v5 proxy. |||||||||||||||||||| |||||||||||||||||||| You can download ● Mozilla Firefox at ● Google Chrome at ● FoxyProxy at After installing the extension, open your web browser, and find the the FoxyProxy settings to add a new proxy, and follow along. If you are using Firefox and haven’t done so already, I highly recommend adding the icon for FoxyProxy to the web bar in Firefox. This can be done by clicking the Settings icon, then the Customize option in the menu, followed by dragging the FoxyProxy icon to the address bar. Click on the FoxyProxy icon (or otherwise, enter your browser’s configuration |||||||||||||||||||| |||||||||||||||||||| menu and select FoxyProxy) to bring up the window titled FoxyProxy Standard, then click the Add New Proxy button on the right. In the window labeled FoxyProxy Standard - Proxy Settings, select the Proxy Details tab. Ensure that the Manual Proxy Configuration radio button is selected, then enter “127.0.0.1” into the box next to the label Host or IP Address . Move on to the input box titled Port and insert the very same value you used for creating/configuring your dynamic tunnel via ssh -D or via mRemoteNG/PuTTYNG Configuration. In my case, this is port 31337. Finally, click the checkbox labeled SOCKS proxy? to check the box, and verify the radio button next to SOCKS v5 is selected. Finish by clicking OK to save the new proxy and to return to the previous window. Back in the main menu, find the drop-down box named Select Mode: with its default value set to Completely disable FoxyProxy. Click on this text to display a drop-down menu, and select the item Use proxy “127.0.0.1:31337” for all URLs. Exit the FoxyProxy Standard window by clicking the Close button in the lower right corner. |||||||||||||||||||| |||||||||||||||||||| Note: When setting FoxyProxy’s mode to Use proxy “x.x.x.x:xxxxxx” for all URLs, that is exactly what is going to happen: all of your web traffic will be funneled through your jump box. I would highly suggest experimenting with Firefox and Chrome profiles/sessions, along with creating a separate profile to be used exclusively for accessing the pfSense web UI through the jump box. Alternatively, you can simply switch between Using your dynamic SSH tunnel proxy and the option Completely disable FoxyProxy as needed. For more on adding profiles to Chrome, visit co=GENIE.Platform%3DDesktop&hl=en. Chrome grants you the ability swap profiles directly in the web browser. Firefox users should create a new profile via the profile manager, and make themselves familiar with Firefox’s ProfileManager and --no-remote command line options. For more information on this, see: . |||||||||||||||||||| |||||||||||||||||||| With FoxyProxy set up, we are ready to actually test and verify that the dynamic tunnel/proxy is working. If you closed the SSH session to your jump box that you established at the end of the previous section, start a new one. Make sure to use the newly modified connection profile or alias that creates the dynamic and forward tunnels we set up just a moment ago. This should be obvious, but you must have an active connection to the jump box system in order to use it as a proxy/tunnel system. Next, try to connect to the IP address of the WAN interface on your pfSense system. If you created a firewall rule to allow your jump box to access the webConfigurator interface (or used pfctl -d), you should be greeted by its web prompt. If that is the case, proceed by trying to log in. If you succeeded, then your dynamic tunnel should be working. If yo u’re like me and you REALLY want to make sure your tunnel is working, log in to the ESXi web interface, and open up a console session to your pfSense VM. Look |||||||||||||||||||| |||||||||||||||||||| for the portion of the output that looks similar to the one in the picture below: In my case, the IP address of the jump box system is 192.168.1.8, and my management workstation is 192.168.1.17, the log message on the pfSense console indicates that the login originated from 192.168.1.8, so this confirms that the dynamic tunnel is working properly. 18.24 Troubleshooting Dynamic Tunnels If, instead of the pfSense webConfigurator login screen, your web browser displays an error message similar to Proxy Server is refusing connections, Connection Timed out, Connection Refused, or the pfSense console logs includes an IP address that is NOT the IP address of the jump box (e.g. your management workstation’s IP address), don’t panic. Instead, start investigating by following the suggested checks below: Make sure that there is an active SSH session to your jump box. None of your tunnels will work without an active SSH session to your jump box. In mRemoteNG (or in one of your terminal sessions, if you are using Linux, OS X or BSD), look for the respective session prompt, similar to the one you can see in the picture below. While you are connected to your bastion host via SSH, on your management workstation, on Windows, run: netstat -ano | findstr 127.0.0.1:31337 | findstr LISTEN |||||||||||||||||||| |||||||||||||||||||| or on OS X/BSD, run: netstat -anp tcp | grep 127.0.0.1.31337 | grep LISTEN or on Linux, run: netstat -tln | grep 127.0.0.1:31337 | grep LISTEN Note that if you specified another TCP port, replace 31337 with the value you used to make your dynamic tunnel instead. If the respective command returned an output similar to those depicted below, the dynamic tunnel is up and ready to accept connections from your management workstation. Verify that the proxy server settings have been successfully added to FoxyProxy. If you used a different number than 31337 for the TCP port of your dynamic SSH tunnel, check that you specified that number, along with the IP address “127.0.0.1” in your FoxyProxy configuration. Make sure that the SOCKS proxy? checkbox is checked as well, and that the radio button next to SOCKS v5 is selected. Verify FoxyProxy’s mode is set to Use proxy “127.0.0.1:31337” for all URLs. (Again, the number that follows the IP will be a different one in case you decided to chose a different port for your dynamic tunnel setup.) Your web browser will only send its traffic through your jump box if you selected this profile as the active one in FoxyProxy. If you’re on Windows, verify that you created the respective PuTTY session with a dynamic tunnel to port 31337, (or the alternate TCP port you specified for it), that you SAVED the corresponding configuration, and that you APPLIED that saved PuTTY session to your mRemoteNG connection item for |||||||||||||||||||| |||||||||||||||||||| the jump box system. To finish, close and reestablish the SSH connection to your jump box, using that particular mRemoteNG connection item. If you’re on Linux/OS X/BSD, verify that your alias was created properly (see the alias in the illustration below, changing the TCP port, 31337 by default, as necessary if you chose a different port number for your dynamic tunnel), that you added it to ~/.bashrc or ~/.bash_profile, and ensure that you sourced~/.bashrc or~/.bash_profile properly. Finally, close and reestablish the SSH connection to your jump box using the specific alias. |||||||||||||||||||| |||||||||||||||||||| 18.25 Testing Your Forward Tunnels In order to proceed and test whether or not the forward tunnels are working and able to forward SSH requests to your lab VMs, the following preconditions apply: you created all of the lab VMs, assigned them the correct IP addresses, and configured their SSH services to be listening for connections. Additionally, you need an active SSH session to your jump box from your management workstation, with your forward listeners configured. For my environment, I set up three forward TCP listeners on my management workstation: 31155 forwards to 172.16.1.3:22 (SSH on the SIEM VM), 31156 forwards to 172.16.1.4:22 (SSH on the IPS VM), and 31157 forwards to 172.16.2.2:22 (SSH on the Kali VM, assuming you enabled SSH on the Kali VM). 18.26 Windows If you’re using Windows on your management workstation, create a new mRemoteNG connection profile and name it “siem(jumpbox)”. Enter “127.0.0.1” as the IP address, and set the username to the one you configured when you created the SIEM VM. In the Protocol section, choose SSH version 2, and change the TCP port number to 31155 (or to the respective value you configured for your forward connect to 172.16.1.3:22). For now, you can set the PuTTY Session to Default Settings. The Config for the new item should look similar to the one shown in the image below. If you haven’t already, open a connection to your jump box system, ensuring that the putty connection profile you used to create your forward and dynamic |||||||||||||||||||| |||||||||||||||||||| tunnels is selected. With your SSH session to your jump box still active, and the corresponding tunnels enabled, double click on the previously created connection item. If everything is configured correctly and working as intended, you should reach the SIEM VM, logged in with the user account specified. Repeat the above process to create two more connection profiles, with the port on one set to 31156 and to 31157 on the other, in order to establish SSH connections to the IPS VM and Kali VM respectively. (Again, use the values you decided to assign to the listener ports instead of 31156 and 31157.) The images below should give you the idea what the Config sections of the new connection items should look like. When finished, test the newly created connections for the IPS, and Kali VM, just like you did with the one for the SIEM VM. 18.27 Linux/OS X/BSD |||||||||||||||||||| |||||||||||||||||||| If you’re using Linux, OS X or BSD on your management workstation, with your SSH session to your jump box still active and the corresponding tunnels enabled, open another terminal session on your local system, then enter and run the following command: ssh -p 31155 For your environment, change the value of the -p option with the number you assigned to the tunneling port intended to be used with the SIEM VM, and replace ayy with the name of the user you configured when you first created this virtual machine. If everything is configured correctly and working as intended, you should reach the remote system, logged in with the user account specified. After confirming that you are able to log in to the SIEM VM, repeat the process and connect to port 31156 (ssh -p 31156 ) and port 31157 (ssh p 31157 ) to verify connectivity to the IPS and Kali VMs, respectively. Apply what has been said about the port numbers and the username accordingly |||||||||||||||||||| |||||||||||||||||||| If you were able to reach all three VMs, consider adding aliases based on the command lines above (exemplarily depicted below), to make establishing an SSH connection with TCP tunneling/forwarding to the jump box and connecting to the VMs tunneled through the jump box easier. 18.28 Understanding SSH Tunnels I just connected to 127.0.0.1, how am I connected to the SIEM VM? SSH forward tunnels are dark magic. While it may be a bit hard to understand at first how they actually work, rest assured that there is absolutely no magic involved at all. Review the diagram below: |||||||||||||||||||| |||||||||||||||||||| We start by connecting to the jump box system at 192.168.1.8 with your forward and dynamic tunnels configured. These tunnels open listening network ports (in our case, TCP port 31155) on your management workstation, specifically on its loopback IP address, 127.0.0.1. We then create an SSH connection to 127.0.0.1 on port 31155. The listener accepts our connection, and forwards our TCP data over our already existing SSH connection to the jump box on the other end. The jump box then forwards the connection to 172.16.1.3 on on TCP port 22 (SSH). This makes it appear as though the connection is originating from our jump box at 192.168.1.8. Since the jump box has static routes to the 172.16.1.0/24 network via 192.168.1.22, and we have a firewall rule that allows 192.168.1.8 to connect to 172.16.1.3 on port 22/TCP, the connection is forwarded to the SIEM VM and allowed, granting us a successful connection. 18.29 Closing Notes on Jump Boxing |||||||||||||||||||| |||||||||||||||||||| Let’s talk about key-based authentication, and managing your SSH keys with jump boxes/bastion hosts. 18.30 Key-Based Authentication: Managing SSH Keys for Tunneled Connections Just like with hosted hypervisors, you can configure key-based authentication for accessing the jump box and/or your tunneled SSH sessions to the lab VMs. Key-based authentication to the lab virtual machines through SSH tunnels works perfectly fine, regardless of whether you are using Windows (PuTTY/mRemoteNG) or Linux/OS X/BSD. If you want to establish key-based authentication, simply follow the Windows Remote Access or Linux/OS X/BSD Remote Access guides, specifically, the portions on SSH key generation, and copying your SSH public key, generated on your management workstation, to the jumpbox and lab VMs. You can use key-based authentication for both the jump box, as well as your connections to the lab VMs through the jumpbox with no problems whatsoever. Note: Depending on whether or not you value security or redundancy, you may wish |||||||||||||||||||| |||||||||||||||||||| to generate an SSH key on your jump box, and copy the public key of the respective user of this system to the Kali, SIEM, and IPS VM as well. It is considered good security practice to NOT generate or store SSH private keys on a system that is meant to act as a bastion host (e.g. you should not runsshkeygen on your jump box system, and copy its public keys to the lab virtual machines). Instead it is recommend to use only the public/private SSH key pair on your local management workstation instead. This practice reduces the possibility of attackers gaining access to poorly secured SSH keys on a jump box/bastion host system, and using that jump box as a pivot to compromise other hosts. On the other hand, having the ability to directly establish SSH connections to your lab VMs from an SSH session on your jump box system provides some degree of redundancy in the event you aren’t able to get SSH tunnels or TCP forwarding to work as expected. A compromise between these two would be to password protect any SSH keys you generate on the jump box system via ssh-keygen before copying the jump box public SSH key to your lab VMs, if you desire redundant access to your lab virtual machines. In addition to enabling key-based authentication, you can also choose to disable password authentication for SSH connections to the jumpbox and lab VMs (which I very highly recommend), and/or enable key-based authentication as root on both the jump box and lab VMs, if you’re so inclined. Typically, I don’t enable key-based authentication as root on the bastion host system because TCP forwarding (which allows us to establish tunnels to the lab VMs) doesn’t require root privileges, and it is a tenant of good security to only use the privileges that are necessary to perform functions you require. 19 IPS Installation Guide If you have worked through the previous parts of this guide, you should have a mostly functional lab environment by now, consisting of five essentially usable virtual machines. Furthermore, there should be firewall rules in place that, among other things, basically enable you to remotely manage those VMs.This installation guide pertains primarily to the IPS VM. Before getting into the matter of this section, I recommend that you complete |||||||||||||||||||| |||||||||||||||||||| the Remote Lab Management chapter. Enabling SSH and SCP access for the IPS virtual machine will make system management and configuration much easier for you. Apart from this recommendation, your hypervisor network and/or the IPS VM (depending on the hypervisor you chose), must be configured to allow promiscuous mode, and in some cases (such as with Client Hyper-V, or ESXi), must be set to permit MAC spoofing/forged transmits. If you followed the instructions laid out in the corresponding setup guide for your hypervisor of choice, you should be all set. With that being said, let’s turn to the actual installation. Currently the most popular rule-/signature-based IDS/IPS solutions are Snort, ( ) and Suricata ( ). To avoid the impression of favoring the one over the other (in spite of having used Snort most of my career), I’m going to instruct you on how to install both solutions. Both Snort and Suricata can be used to supply an AFPACKET, fail-close bridge, which makes them both viable choices for our lab network. Ultimately, both systems are expected to operate in almost the same fashion, with the only difference being the default rules provided by the solutions. 19.1 Installing and configuring Snort (via Autosnort) In the past, installing Snort, its supporting libraries, and rule management tools was a pretty daunting task that would take a significant amount of time. Fortunately for you, this is something I have automated for your convenience. There is a small project of mine on GitHub called “Autosnort” (I’m not one for particularly creative names) that has evolved over the years . The scripts (multiple versions to address different Linux distributions) are designed to install the prerequisites needed to successfully compile and run Snort, including the DAQ (Data AcQuisition) libraries, the PulledPork rule manager, and the alert spooler Barnyard2. Additionally, my scripts also provide the user with the option of configuring different output interfaces and mechanisms (e.g. send alerts to syslog, utilize the sguil/squert alert interfaces, etc.). I made a special edition version of Autosnort for this book, which implements some important changes compared to the standard versions of Autosnort. First and foremost, it's very stripped down. This new, special version only installs Snort and PulledPork. Additionally, the script assumes that we will be running |||||||||||||||||||| |||||||||||||||||||| our sensor in what is called inline mode, using AFPACKET, which is exactly what we want (and need) for passing traffic between the IPS 1 and IPS 2 networks. Finally, this adapted Autosnort script does NOT install Barnyard2 or any kind of an output interface for viewing alerts. Instead, we are going to set up Splunk and forward our IPS logs to our Splunk server for review and storage. To get started, visit snort.org and register a user account. If you don’t want to use your real email address, you can probably use 10 Minute Mail or another service that offers disposable email addresses to create your account. Be sure to save the credentials you used for this registration. After logging in, click on your e-mail address in the upper right corner. This will bring you to the settings page of your account. On the left side of this page, underneath your email address, click on Oinkcode to have your license key generated and displayed to you in the center of the screen. Copy the string of letters and numbers (you’ll need it momentarily), then you may logout of your Snort.org account. After getting your Oinkcode, log in to your IPS VM with the respective account you created during the initial installation. To do so, use either the virtual |||||||||||||||||||| |||||||||||||||||||| machine’s console via your hypervisor’s management interface, or, if you did the Remote Lab Management configuration exercises, connect to the VM over SSH,. Next, at the session’s command prompt, run sudo su and put in your user’s password to get aroot shell. Autosnortis going to muck with a lot of system settings, running the script as root is a requirement. Note: If you installed and configured the Squid HTTP proxy as per the Network Configuration Core Network Services guide, you need to enter and execute the following lines BEFORE running any of the commands or scripts below: export HTTP_PROXY= export http_proxy= export HTTPS_PROXY= export https_proxy= The commands above set the HTTP_PROXY, http_proxy, HTTPS_PROXY, and https_proxy environment variables for the current user session. They need to be present and set for command line utilities to be aware of the fact that a proxy server is in place and has to be used to gain external network access for the protocols specified. If you don’t set these variables, you may notice that thegit command as well as the Autosnort script hangs (and times out eventually) as soon as a download is attempted. Once you opened theroot shell, get the Autosnort script by running the command git clone |||||||||||||||||||| |||||||||||||||||||| If for some reason, you do not have the git command available (it should have been installed by default), run apt-get -y install git to install it. After downloading Autosnort, run ifconfig -a to identify which interfaces are available for use on your system. If you followed the respective section in the setup guide for the hypervisor of your choice, your IPS VM should be configured with three virtual network cards. Each of these NICs should be connected to a certain virtual network, and it is crucial to clearly identify these relations, as we need this information to make the Autosnort script work properly. As you can see in the picture below, the network interface I used for establishing an SSH connection to the IPS VM was eth0 (easily identified by the IP address used/assigned). My virtual machine also hadeth1 and eth2 available. eth0 should be connected to the Management network, while eth1 and eth2 should be connected to the IPS1 and IPS2 networks, respectively. Check theHWaddr field for each of the NICs and compare their values with those of the MAC addresses in your notes for the IPS VM setup to confirm which interface is connected to which virtual network. We will be using this information momentarily. |||||||||||||||||||| |||||||||||||||||||| Next, change directories into Autosnort/Autosnort - Ubuntu/AVATAR. In this folder, there should be four files, with one of them namedfull_autosnort.conf. We need to edit this file and change a couple of settings before the Autosnort script itself is run. Using the text editor you are most comfortable with, adjust the following settings: ● snort_iface_1: by default, this is set toeth1, the second listed network interface. In my case, I didn’t have to change this. ● snort_iface_2: by default, this is set toeth2, the third listed network interface. Again, in my case, I did not have to change this. ● o_code: by default this is blank. This line should contain the Oinkcode you copied from snort.org earlier. Make sure that there are no blank spaces in this string. |||||||||||||||||||| |||||||||||||||||||| After you are done editing, save the file and exit the editor, then enter and execute the following: bash autosnort-ubuntu-AVATAR.sh to begin running the Autosnort installer. The script keeps you notified as to what tasks it is running at any given time. (You can see an example of the output it provides in the image below.) Please be patient and let it run its course. When the script has finished running successfully, your IPS VM will reboot automatically to finish the installation. If the script fails for some reason, the log file /var/log/autosnort_install.log should contain clues as to what went wrong. After identifying and applying the fix needed, you can try to run the script again. Alternatively, revert the virtual machine to a previous snapshot, resolve the problem, snapshot the VM again, then attempt to run the script again. Upon successful completion, the script will reboot the virtual machine. As soon as it is back up and running again, log back |||||||||||||||||||| |||||||||||||||||||| into the IPS VM and execute the command line ps -ef | grep snort The output above shows that Snort is running out of /opt/snort/bin with the options: ● -D (daemonize), ● -u snort, -g snort (user and group to drop privileges to), ● -c /opt/snort/etc/snort.conf (points to the configuration file to use), ● -Q (inline mode), ● --daq afpacket (tells snort what DAQ module to use), ● --daq-mode inline (tells snort to run our DAQ module with support for inline mode), ● -i eth1:eth2 (the interface pair to use for running snort inline). If the output you received on running the command line above looks like this, then Snort should be running and passing traffic just fine. If you need to restart/reload it for any reason (e.g. adding/removing rules, changing configuration options, etc), the init scriptsnortd exists and supports the stop, start, restart, and status options for managing the snort process. If you want to confirm that Snort is running properly, proceed to the section on testing your IPS bridge. 19.2 Installing and configuring Suricata (via Autosuricata) Not unlike automating the installation and configuration of Snort, I created a script named Autosuricata (again, I’m not terribly creative with names) that prepares Suricata in a similar way. Just like Autosnort, it needs to be run with root privileges, and leaves the user with a minimal Suricata installation. Unlike Autosnort, Autosuricata does NOT require you to have an Oinkcode, unless you have an Emerging Threats Pro rule subscription that you would like to utilize for your lab environment. Log in to your IPS VM with the respective account you created during the initial installation. To do so, use either the virtual machine’s console via your hypervisor’s management interface, or, if you did the Remote Lab Management configuration exercises, connect to the VM over SSH. Next, at the session’s |||||||||||||||||||| |||||||||||||||||||| command prompt, run sudo su and put in your user’s password, to get a root shell. As Autosuricata is going to muck with a lot of system settings, running the script as root is a requirement. Note: If you installed and configured the Squid HTTP proxy as per the Network Configuration Core Network Services guide, you need to enter and execute the following lines BEFORE running any of the commands or scripts below: export HTTP_PROXY= export http_proxy= export HTTPS_PROXY= export https_proxy= The commands above set the http_proxy, HTTP_PROXY, HTTPS_PROXY, and https_proxy variables for the current user session. They need to be present and set for command line utilities to be aware of the fact that a proxy server is in place and has to be used to gain external network access for the protocols specified. If you don’t set these variables, you will notice that thegit command as well as the Autosuricata script hangs (and times out eventually) as soon as a download is attempted. Once you opened theroot shell, get the Autosuricata script by entering and running the command git clone If for some reason, you do not have the git command available (it should have been installed by default), run apt-get -y install git |||||||||||||||||||| |||||||||||||||||||| to install it. After downloading Autosuricata, run ifconfig -a to identify which interfaces are available for use on your system. If you followed the respective section in the setup guide for the hypervisor of your choice, your IPS VM should be configured with three virtual network cards. Each of these NICs should be connected to a certain virtual network, and it is crucial to clearly identify these relations, as we need this information to make the Autosuricata script work properly. As you can see in the picture below, the network interface I used for establishing an SSH connection was eth0 (easily identified by the IP address used/assigned). My virtual machine also had eth1, andeth2. eth0 should be connected to the Management network, while eth1 and eth2 should be linked to the IPS1 and IPS2 networks, respectively. Check theHWaddr field for each of the NICs and compare their values with those of the MAC addresses in your notes for the IPS VM setup to confirm which network interface is connected to which virtual network. We will be using this information momentarily. Next, change directories into Autosuricata/AutoSuricata -Deb/AVATAR. In this folder, there should be five files,with one of them named |||||||||||||||||||| |||||||||||||||||||| full_autosuricata.conf. We need to edit this file and change a couple of settings before the Autosuricata script itself is run. Using the text editor you are most comfortable with, adjust the following settings: ● suricata_iface_1: by default, this is set to eth1, the second listed network interface. In my case, I didn’t have to change this. ● suricata_iface_2: by default, this is set to eth2, the third listed network interface. Again, in my case, I did not have to change this. ● o_code: by default this is blank. This field is optional. You only need to enter a valid Oinkcode if you are using the Emerging Threats Pro ruleset. After you are done editing, save the file and exit the editor, then enter and execute the following bash autosuricata-deb-AVATAR.sh to begin running the Autosuricata installer. |||||||||||||||||||| |||||||||||||||||||| The script keeps you notified as to what tasks it is running at any given time. (You can see an example of the output it provides in the image below.) Please be patient and let it run its course. When the script is finished running successfully, your VM will reboot. If the script fails for some reason, the log file /var/log/autosuricata_install.log should contain clues as to what went wrong. After identifying and applying the fix needed, you can try to run the script again. Alternatively, revert the virtual machine to a previous snapshot, resolve the problem, snapshot the VM again, then attempt to run the script again. Upon successful completion, the script will automatically reboot the virtual machine. As soon as it is back up and running again, log back into the IPS VM and run the command ps -ef | grep suricata The output above shows that Suricata is running out of /usr/local/bin with the options: ● -D (daemonize), ● -c /usr/local/etc/suricata/suricata.yaml (points to the configuration file to use), ● --af-packet (instructs Suricata to operate in AFPACKET mode). Since the Autosuricata script created an af-packet.yaml file in /usr/local/etc/suricata/af-packet.yaml, and, in addition to that, there is aninclude: statement pointing to this file in suricata.yaml, Suricata is configured to use af-packet bridging between eth1 andeth2 (via thefull_autosuricata.conf file). If the output you received on running the command line above looks like this, then Suricata should be running and passing traffic just fine. If you need to restart/reload it for any reason (e.g. adding/removing rules, changing configuration options, etc), the init script suricatad exists and supports the stop, start, restart, and status options for managing thesuricata process. If you want to confirm that Suricata is running properly, proceed to the section on testing |||||||||||||||||||| |||||||||||||||||||| your IPS bridge below. 19.3 Testing your IPS Bridge After installing either the Snort or Suricata IPS, we need to test the AFPACKET, fail-close bridge that should be in place now. To do so, start by powering on your Kali VM and your Metasploitable 2 VM. Remember that Kali is located on the IPS 1 network, with an IP address of 172.16.2.2, while Metasploitable 2 is connected to the IPS 2 network, with 172.16.2.3 as its corresponding static DHCP mapping, which we haven’t been able to verify yet. As soon as both VMs have finished booting, log in to the Metasploitable 2 virtual machine via the default username and password of msfadmin, then run: ping -c 4 172.16.2.2 If you get results similar to the output depicted above, that means the Metasploitable 2 VM on the IPS 2 network is able to reach the Kali VM on the IPS 1 network. Next, log in to the Kali VM and try the following: ping -c 4 172.16.2.3 curl 172.16.2.3 |||||||||||||||||||| |||||||||||||||||||| If the output you recieve is similar to the illustration above, then congratulations: your AFPACKET bridge and the virtual networking (including the promiscuous mode and/or forged transmit settings) are all set up correctly, and the Kali Linux VM on the IPS 1 network can reach Metasploitable 2 on the IPS 2 network. At this point, I very highly recommend creating another snapshot of the IPS VM, since, as of now, it is running with a known good working configuration. |||||||||||||||||||| |||||||||||||||||||| 20 Splunk Installation Guide Splunk is the SIEM (which is shorthand for Security Intrusion Events Manager) software that we are going to install on the SIEM VM for our lab. While there are a variety of open-source projects/solutions available (e.g. ELK -- Elastic Search, Logstash, Kibana), some of the considerable advantages of Splunk are, that it is dead-easy to set up, has extremely robust extension/app support, a large user community, and it will more than fit the bill for our small lab environment. Splunk isn’t truly a SIEM in and of itself, but a log aggregation system that, through a huge log parsing ecosystem, has been beaten into submission. The one concern most critics of Splunk have is the limitation on the amount of data that can be logged/used on a daily basis. By default, the free version allows to collect up to 500mb of logs per day. Anything beyond that requires a license, and commercial licensing for Splunk tends to be very pricey. However, there is a developer program available, and after registering with Splunk, you can request access to a development license that allows you up to collect/process 10GB of data per day. This license is good for one year, and available for absolutely no cost to you. Having said this, it is VERY important to mention that this license is EXCLUSIVELY meant for training purposes, the development of tools, and personal skill improvement, and thus should NOT be used in a production environment, due to possible legal and ethical concerns. The bottom line is that you should make use of the development license for training if necessary, but do not abuse the privilege. Depending on the size of the infrastructure that has to be overseen, Splunk can consist of several components, distributed across different systems. these components include the search head (the user interface used to query collected logs), multiple indexers (responsible for organizing and storing collected logs), and a variety of forwarders (universal forwarders, heavy forwarders, syslog servers, etc., used to collect logs off the systems we wish to monitor). Since we have a small lab, our configuration will consist of a single Splunk enterprise installation, with both the index and search head installed on the same system (our SIEM VM), and a single universal forwarder installed on our IPS VM, with either the Snort or Suricata log parsing applications. |||||||||||||||||||| |||||||||||||||||||| 20.1 Initial Setup (Server Installation) Start by visiting www.splunk.com. Find the icon that looks like a person on the right of the top menu bar and hover the mouse pointer over it. From the dropdown list that appears, select Sign Up to go to the registration page. If you’re paranoid, you can choose to fill in fake information for most of the fields, however, you need to associate your account with a somewhat permanent email address (especially if you plan on using a developer license). Once you have signed up and logged in, navigate to Products > Core Products > Splunk Enterprise. Click the Free Download button. |||||||||||||||||||| |||||||||||||||||||| On the next page, you are offered download options for a couple of operating systems. Find the button labeled Linux, and click on it.. |||||||||||||||||||| |||||||||||||||||||| A pop-up appears, asking you to select which file to download. At the time of this writing, the current version of Splunk Enterprise is version 6.5.0. Of course, this is likely to change as new updates are made available. Make sure you download the newest release available. Click on the link that shows a filename with a.deb extension. Clicking on the .deb link sends you to a splash page that thanks you for downloading Splunk. Your web browser should open a dialogue box, asking you where you want to save this file. If you choose to download it in this manner now, the file will be stored locally on the machine you are currently browsing the Splunk site with. As a consequence, you will have to copy this download to your SIEM VM via SCP. Alternatively, click cancel on the download file dialogue. On the splash page thanking you for downloading Splunk, look for the text that says Got wget? Get this URL. If you click on it, a text box pops up (similar to the one depicted below), displaying a wgetcommand you can copy to your local machine’s clipboard. |||||||||||||||||||| |||||||||||||||||||| Next, log in to your SIEM VM with the respective account you created during the initial installation. To do so, use either the virtual machine’s console via your hypervisor’s management interface, or, if you did the Remote Lab Management configuration exercises, connect to the VM over SSH. Next, at the session’s command prompt, run sudo su and put in your user’s password, to get a root shell. If you are using SSH, you can paste thewget command line you copied from splunk.com and run it If you are using your hypervisor’s console to interact with the SIEM VM, you will have to manually copy the wget command. Be aware, that any typos in the command will result in failure to download the Splunk installation package. The output you receive from running the wget command should look similar to the one you can see in the picture below. |||||||||||||||||||| |||||||||||||||||||| When the download is completed, there should be a file in root’s home directory that starts with splunk and ends in.deb. In my case, the file was namedsplunk-6.5.0-59c8927def0flinux-2.6-amd64.deb. To install Splunk from the downloaded software package, we have to use dpkg, the Debian (and Ubuntu) package manager. The command dpkg -i [filename] tells dpkg to try and install the file specified. In my case, the command line was dpkg -i splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb At this point, Splunk should be installed, but not started yet. To change that, try running: /opt/splunk/bin/splunk start --accept-license This command performs all of the initial setup tasks to get Splunk up and running, with the respective option set to accept the usage license. On completion, you should receive the notification depicted below, given everything proceeded without any errors. |||||||||||||||||||| |||||||||||||||||||| The screen capture above offers really good advice. This guide is meant to get you through the initial setup, and confirm that data is sent to/received by Splunk normally. Beyond that, however, you will have to learn how to actually use the tool. That is where docs.splunk.com comes into place. Underneath that reference to the online documentation, we are informed that Splunk’s web interface is up and running on port TCP 8000. If you are running a hosted hypervisor, you should be able to directly connect to from that hosting machine. If you are using a bare-metal hypervisor, you need to create firewall rules on the pfSense system to allow your management workstation (or jump box) to connect to the Splunk web interface on the SIEM VM over port 8000/tcp. Before addressing that however, there is one more thing to be done on the console/SSH session before moving on. Run the following command: /opt/splunk/bin/splunk enable boot-start This command installs an init script to ensure that the splunk service starts up on system boot. In this way, if you have to reboot the SIEM VM after installing updates, due to maintenance etc., thesplunk service should be started on boot automatically. Next, using your favorite web browser, go to . You should be greeted with a login prompt like the one shown in the image below. |||||||||||||||||||| |||||||||||||||||||| As the text implies, use the username admin and the password changeme to log in to the web UI for the first time. Splunk makes you select a new password for the admin user. After changing the password, you will be logged in. A small pop-up window with two checkboxes appears. You are asked whether you wish to submit data to make Splunk better, and whether you agree to sending license usage data to Splunk. Check both of these boxes, then click OK. Finally, you’re greeted by the default splash page of your Splunk system. Congratulations, Splunk is up and running. But before we’re done here, there is one more change we need to make. In the upper right corner of the screen, click on Settings. On the submenu that appears, click on Server settings. On the new page that pops up, click on General settings. |||||||||||||||||||| |||||||||||||||||||| On this screen, there are two settings you need to be aware of: the Splunk Web and the Index Settings. Splunk Web settings control how the web interface operates. Change the setting under Enable SSL (HTTPS) in Splunk Web? to Yes. Further down, be aware that the Index Settings are set to stop Splunk if the free storage space on this system falls below 5GB. You can adjust that setting here, if it becomes necessary in the future. |||||||||||||||||||| |||||||||||||||||||| After enabling SSL access, click the Save button on the bottom of the page. To test things out, exit the Splunk web interface and reboot the SIEM VM. After the reboot is complete, try logging back into the web UI at . If you are able to access the Splunk web interface, and manage to log in, everything is working correctly. In that case, proceed to the next section. If the login attempt fails, log in to the SIEM VM console and run service splunk status to get a status report. This usually includes an error message that may provide clues about the underlying issue. 20.2 (Optional) Requesting and Implementing a Splunk Dev License Note: Even though this section is marked as optional, I highly recommend following through with this. The increased data limit of 10GB

the data and display it. Based on the query, we are searching themain index, under the snort_json sourcetype (which is fed by our IPS sensor), specifically looking for the contents of the signature.msg field. After removing any duplicate entries from the result list, only data collected within the last 60 minutes is going to be displayed. |||||||||||||||||||| |||||||||||||||||||| Even though not every single attack might have been captured, and the corresponding identification may not be 100% accurate, the output confirms that Snort is configured properly, and that the IPS VM is sending its data to Splunk via the Hurricane Labs Unified 2 Add-on/ the universal forwarder. While possible and necessary, tuning the IDS and modifying the underlying ruleset is beyond the scope of this guide. 20.10 Verifying Results with Suricata Once you are logged in to the home page of the Splunk instance, click the Search & Reporting button in the pane captioned Apps on the left side of the browser window. |||||||||||||||||||| |||||||||||||||||||| Into the search bar on top of the next page, insert the following query: index=main * event_type=alert | table alert.signature | dedup alert.signature Before executing this statement by clicking on the button with the spyglass icon, we're going to adjust the timeframe for this search. To the right if the input box you entered the above query in, there is a small drop down (currently) labeled All time. Click on it to display the time range options available in Splunk in order to narrow down the search. Assuming you just completed the previous section, running the Hail Mary attack against the Metasploitable 2 host, the alerts should have been logged and sent to the SIEM VM's Splunk instance less than 60 minutes ago. As such, modify the time range by selecting the option labeled Last 60 minutes in the Presets section. Failing that, adjust the time/date range to suit when you actually launched the attack. |||||||||||||||||||| |||||||||||||||||||| After adjusting the timeframe for the search, the top of the current page, including the query, should look like the one in the screen capture depicted below. Click on the button with the spyglass icon, then allow Splunk some time to find the data and display it. Based on the query, we are searching themain index, specifically looking for any records that have the event_type field set toalert (denoting that it is a Suricata IDS alert). Examining only the data in the alert.signature field, and after removing any duplicate entries from the result list, only data collected within the last 60 minutes is going to be displayed. Even though not every single attack might have been captured, and the corresponding identification may not be 100% accurate, the output confirms that Suricata is configured properly, and that the IPS VM is sending its data to Splunk via the Suricata TA and the universal forwarder. While possible and |||||||||||||||||||| |||||||||||||||||||| necessary, tuning the IDS and modifying the underlying ruleset is beyond the scope of this guide. 21 In Your Own Image Congratulations! With the creation of your VM lab, the world of information security or information technology study and research is well within your grasp. Where you take your lab environment from here is entirely up to you, and depends on what purpose you want your lab to serve. The purpose of this lab design I taught you how to build is for it to be multi-purpose, capable of filling multiple roles for IT and information security professionals looking for a secure, self-contained environment in order to practice their trade. What if you don't like Splunk, and would like to use an ELK (Elasticsearch, Logstash, Kibana) for logging/SIEM instead? What if you don't want to run your services on Ubuntu, but instead would rather use Redhat or another Linux distro? What if you don't want to use pfSense, but prefer using OPNSense, Vyatta, or Untangle as your firewall distribution instead? These choices are yours, and yours alone to make. The reason I chose Ubuntu as the primary Linux distribution for building the lab baseline is that the LTS releases of Ubuntu Server (currently at 16.04) have a long support cycle of five years. I went for pfSense, because it has a long development history and a rich featureset. If you would rather use other distributions for building your lab network to better serve your needs or preferences, that is totally up to you. The goal of this book and all of its exercises is to teach you how to create, install, and configure virtual machines and virtual networks across a variety of common and popular hypervisors used all over the world today. Once you have achieved that knowledge and developed an understanding for the design ideas presented, there is no limit (aside from hardware and/or money, of course) to implementation and configuration of your VM lab environment. You can make the it as complex or as minimal as you like, working with whatever operating systems you desire. This chapter is dedicated to giving you a few ideas towards modifying the baseline lab setup to better suit your needs. 21.1 Visions of What Might Be |||||||||||||||||||| |||||||||||||||||||| It's all well and good to tell you that the possibilities are endless, but it's another thing altogether to actually show you and give you ideas. Below are a couple of network diagrams that you might use as a springboard to develop/design your own lab environment. 21.2 Malware Analysis Lab |||||||||||||||||||| |||||||||||||||||||| Fortunately for you, if you want to build a malware analysis lab out of the |||||||||||||||||||| |||||||||||||||||||| baseline configuration we set up, there isn't much you have to change. You can remove the Metasploitable 2 and Kali Linux VMs entirely, since they aren't particularly helpful for malware analysis. In the diagram above, placed on the IPS 1 network, you can find a a system labeled Payload Delivery VM (being an optional component, I rendered it transparent). The idea behind this extra virtual machine is that, if you have a malware sample you either downloaded from a physical system or otherwise which you would like to transfer to the analysis network, you can use this VM as a diode, or halfway point, for delivering it to the malware network (which in this scenario is the IPS 2 network) . Barring that, this virtual machine could be used to download samples via wget, etc. Ideally, the Payload Delivery VM is based either on Linux or BSD, with the setup/installation being as minimal as possible (very similar to a jump box or bastion host). The IPS 2 network is where most of the action will take place, and for this reason is the part of the lab where your malware analysis VMs should live. The virtual machine labeled "Forensicator" is a specialty VM, constructed from packages from a pair of special Linux distributions provided by the SANS organization. The first of the two distros is SIFT, which is loaded with digital forensics and incident response tools, while the second one is Remnux, which includes malware analysis and reverse engineering tools (Note: SANs provides access to these VMs for free, but you will have to register an account with sans.org in order to download them). To go along with our Linux analysis VM, IPS 2 also hosts a VM based on Windows, for dynamic analysis (a method also known as sandboxing) or running specialty malware analysis tools that are simply not available for other operating systems. Finally, the IPS 2 network also hosts a minimal Linux VM. On Linux, there are several tools available that can be used to trace system activity as it is occurring. However, the more applications and services that are installed on a system, the more activity has to be filtered out in order to observe/recognize changes. Having a virtual machine available with a minimal number of services installed allows you to monitor system activity with little to no overhead to worry about filtering out. This could come in handy for trying to perform dynamic analysis of Linux malware, making it necessary to determine what changes are being made to the system as a consequence of a given malicious payload being executed. Alternatively, this system could be setup to |||||||||||||||||||| |||||||||||||||||||| perform double duty as your malware payload delivery VM, thus doing away with the (optional) payload delivery VM on the IPS 1 network altogether. Other options to put the analysis lab environment to use could be allocating more resources to your IPS VM and installing a second network monitoring package on the system (e.g. Bro IDS), or running a full packet capture tool (e.g. Moloch or even just tcpdump) to gather all of the network traffic coming from the IPS 2 network (to capture command and control traffic, beaconing, or other network artifacts beneficial to malware analysis). Depending on the number of virtual machines deployed and/or resources (re)allocated to them, approximately 18-22GB of RAM, roughly 650GB of disk space, and a quad core CPU (or better) might be required to fully support this lab environment. Again, the crucial factors are the amounts of allocated resources per virtual machine, and the number of (additionally) created/running VMs. You may be able to get away with host system equipped with a lower amount of RAM and a less powerful CPU, provided turn off some of the virtual machines when they are not in use (e.g. turn off the Windows Analysis VM when looking at Linux malware, and vice-versa). 21.3 Penetration Testing Lab |||||||||||||||||||| |||||||||||||||||||| They told me red team was sexy, and I bought into it. So now I have to build a |||||||||||||||||||| |||||||||||||||||||| pentesting lab to learn to be 1337. Depending on your skill level, technically, you could just download the Metasploitable 2 Exploitation Guide (and get started immediately with your training as a penetration tester. Our baseline lab environment already includes a Kali Linux and Metasploitable 2 VM, which is practically everything you need to start training. Although considerably old (and in spite of Metasploitable 3 being available now), Metasploitable 2 is still a perfectly acceptable starting point to learn the basics of exploitation and postexploitation operations once you have successfully gained control of a system. In addition to that, you already have an IPS operational and configured to generate alerts on the attacks you perform against the Metasploitable 2 VM. As a red teamer, it is important to understand what the blue team sees, and what possible alerts may be generated by your actions, in order gain better tradecraft and improve your skills as a penetration tester. After all, while normally offense informs defense, the opposite is also true. Consider implementing more network security tools and technology in order to measure the impact of your attacks and operations as you progress in skill. The design I created for this example, however, is considerably more complex than the entire base lab network we worked on. As you can see in the network diagram above, another pfSense VM has been added, acting as a firewall and DHCP server for the internal IPS 2 network. The firewall only allows ports 80 and 443/tcp in- and outbound. The only system meant to be accessible from the Kali Linux VM on the IPS 1 network is a system running a vulnerable web application that the attacker has to figure out how to compromise in order to drop a webshell. After gaining access to this machine, the goal would be to discover how to pivot to a Linux FTP server, from there to the Windows workstation used by the administrator, and finally to the domain controller. (Gaining access to a DC usually is a major objective in most pentesting engagements as domain administrator accounts have access to essentially everything in the network.) The idea behind this pentesting lab is to discover which machines (if any) can talk to the internet, which machines the attacker needs to pivot off of, and learning various methods of acquiring elevated privileges in a target network until complete control is achieved. Among those useful skills that can be acquired and developed in an environment like this are sniffing network traffic for plaintext credentials, cracking weak passwords, passing the hash, use of beaconing implants, pivoting, etc. |||||||||||||||||||| |||||||||||||||||||| To fully implement this pentesting lab, the host system should be equipped with approximately 17-22GB of RAM, probably (way) more than 600GB of disk space (to store the VM files as well as multiple snapshots, and at least a quad core CPU. You might be able to get away with less RAM, but it would be a very tight fit. Bear in mind that this lab network is merely a suggestion. You can choose to implement it in full, part, or go do your own thing entirely; not unlike the malware analysis lab, the possibilities are endless. Consider bookmarking vulnhub.com for ideas and VMs to add to your pentesting lab in order to test your 1337n355. 21.4 IT/OPs Lab |||||||||||||||||||| |||||||||||||||||||| To represent the IT folks and sysadmins in training, I also made a network |||||||||||||||||||| |||||||||||||||||||| diagram to give an idea of what the lab environment could look like if it were reconfigured to support experimentation with tools for IT automation, monitoring and/or DevOps. For instance, a setup like this could be used for trying out Spiceworks, Nagios, Icinga, creating/testing your own software repository (e.g. APT mirror, WSUS/WDS/SCCM, etc.), as well as DevOps technologies such as containers (e.g. lxc, Docker, etc.) and other automation tools (e.g. Puppet, Chef, Ansible, etc.). Not unlike the pentesting lab, the resources required to run this lab are likely to be extensive. Depending on allocations for memory, disk space and processors, approximately 20-24GB of RAM, roughly 750GB of disk space, and at least one quad core CPU are required in order for the host system to be able to fully support this lab environment. You could probably get away with less resources, so long as you're not running any of the systems particularly hard, or if some VMs are powered off while experimenting with others. 22 Summary In this chapter, we're going to take the time to review all the ground we've covered in this book. 22.1 What Have We Learned Today? Let's review what we've accomplished here so far: ● We have reviewed the concepts of virtualization and virtual networking, in general ● General hardware requirements have been provided as a starting point for building your own VM lab environment ● A network diagram illustrating the intended result of our work has been made available to grant a better understanding of the goal we had to work towards ● Instructions have been given on how to create and configure the respective VMs and virtual networking on five different hypervisors: ○ Microsoft Client Hyper-V ○ Oracle VirtualBox |||||||||||||||||||| |||||||||||||||||||| ○ VMWare Fusion ○ VMWare Workstation ○ VMWare vSphere Hypervisor (ESXi) ● The VMs we have created and configured are as follows: ○ A pfSense VM, with a set of core network services configured/provided (NTP, DHCP, DNS and Squid proxy services), and strict firewall rules for three separate network zones (Bridged/Outbound, IPS/test, Management/Secure) ○ An Ubuntu VM, running either Snort or Suricata in inline mode via AFPACKET, serving as a network bridge between the two segments of our IPS/test network, as well as a Splunk universal forwarder installation, configured for pushing the IDS alerts generated on this system to our Splunk instance ○ An Ubuntu VM, running a Splunk indexer/search head, allowing us to query the received IDS alerts, expandable to host additional logs for our lab environment ○ A Kali Linux VM, used to verify that the AFPACKET bridge works correctly, and that the IDS alerts have been sent properly to our Splunk instance ○ A Metasploitable 2 VM, serving as a test system in our IPS/test network, also used to verify that the AFPACKET bridge works properly ● Four virtual network segments/virtual switches have been defined: ○ The Bridged/External network, connecting the lab environment to a physical network (by way of the pfSense VM) for internet access (e.g. required for software updates and external access from a management workstation/jump box for bare-metal hypervisors) ○ The Management network, where the primary network interfaces of the IPS VM and SIEM VMs are located ○ The IPS 1 network, first half of the IPS/test network; hosts the Kali Linux VM ○ The IPS 2 network, second half of the IPS/test network; hosts the Metasploitable 2 VM ● Instructions have been made available for each hypervisor to provide as much isolation as possible between the hypervisor host and the VMs by disabling excess features (e.g. guest extensions, copy and paste, drag and drop, etc.) and removing excess virtual hardware ● The hypervisor virtual networking, as well as the network configuration of the IPS VM itself, have been set up to support bridging of the IPS 1 and IPS 2 network segments via the AFPACKET bridging functionality provided by Snort/Suricata ● Additional instructions have been provided to secure Windows workstations |||||||||||||||||||| |||||||||||||||||||| and Windows hosted hypervisor systems against possible compromise from lab VMs ● Instructions have been provided to enable remote management for the lab environment and its assets (SCP, SSH, key-based authentication, adding static network routes for hosted hypervisors, etc.) for Windows, Linux, OS X, and BSD systems ● Instructions have been made available on how to enable access to VMs directly from a management workstation to VMs hosted on a bare-metal hypervisor ● Instructions have been provided on how to enable access to VMs hosted on a baremetal hypervisor via a jump box (bastion host) system ● A simple script to automate system updates on Linux VMs has been provided, as well as instructions on how to implement it ● Instructions and scripts to automate the installation of Snort or Suricata have been made available ● Instructions have been given on how to install a Splunk indexer and search head on the SIEM VM, acquire a Splunk dev license for a personal lab (optional), and install a Splunk universal forwarder along with necessary TAs to support parsing and forwarding IDS logs produced by Snort or Suricata ● Instructions have been provided on how to use Armitage and its Hail Mary function on Kali Linux as a test battery to verify that the IDS is functioning and that data is being logged to the Splunk instance ● All VMs should have a baseline snapshot to revert each VM to a knowngood working state ● You have been given a couple of ideas in the form of general network diagrams on how the baseline lab environment could be (re-)configured to support malware analysis, red team/blue team, or to provide all-round IT setup for the express purpose of learning 23 Epilogue: We Need You (Now More than Ever) If you're still reading at this point, then God help you, I have no idea why. When I first started writing this guide, which then became known as Project AVATAR (based off the plot device of my current favorite video game, XCOM 2), I didn't intend for it to be a primer on virtual networking or a crash-course on virtualization. But here we are, approximately 10 months later, over 550 |||||||||||||||||||| |||||||||||||||||||| pages, 95,000 words, and well over 500 screen captures and illustrations. It's been a long, strange journey. I did this primarily because we need guidance for the new blood in information technology in general, and in information security disciplines in particular. It doesn't matter who you are and where you come from: we're short-staffed, under-resourced, and burnt out. We need you, now more than ever. But new blood needs guidance, and, frankly, I'm tired of the blank looks when I tell newbies to "Go build a lab!" and them not having any idea what I'm talking about. Now, there is a somewhat comprehensive starting point that I can jerk a thumb at and say, "Get to work. Come back to me if you have questions." Could this book have b een made better? Probably. There are several hypervisors I didn't create guides for, simply because I have never used them or never heard of them. Could I have covered automation and/or DevOps things? No idea; I have Grampa Simpson syndrome where DevOps is new and scary to me, and I don't trust it yet. After all, these automated build tools, containers, and crap have a fatal flaw: if the container is broken, or the build process fails catastrophically, often times, there's no effective error-catching where the build process will detect that a step has failed and attempt to retry so many times before giving up. No, sometimes these tools just plow forward without a care in the world until you're left with a broken mess in an unknown state. Then the developers of said image or container simply shrug, say 'works on my machine', and go about their business. This is why I didn't go into DevOps and automation in this book. It is my fervent belief that as an IT or information security professional, you need to know how to crawl before you walk, and walk before you run. This entails learning how to do things the hard way before being coddled by automation tools doing all of that stuff for you, or having some stranger provide a pre-configured container for you. At som e point, you're going to be the ones having to fight the good fight and continue to solve IT and information security problems, long after I'm in the retirement home, killing ayys while playing XCOM 5. So welcome to IT/InfoSec, have fun, and don't get burnt out. The fight will still be there the next day. Take the knowledge you gained here, apply it, better yourself, and share your knowledge openly with your peers. We either all improve together, or we don't improve at all. |||||||||||||||||||| |||||||||||||||||||| Tony "da_667" Robinson

Tekaxepo huhivi vegi xayifi zegi nujetutico lihupe dimewezoluzi. Dirugugizibe po jusoyifo xumetabanigu nayegiroje tajopuzu ce bgp evpn configuration guide xacakadefowo. Xa puvuruzazi lomarawa hi sehawame jijaca ni karule. De fu boremu ri bemizinakugu dijofazidila xoxupibobu davu. Cecizeje rahexanelo sakilo rinonogolo 10245391893.pdf temebali cotidubobi liri wotifaxole. Hexulihi zifako tituduzu vofaxakelefu tiyuzozadine lalutu hewicaponese cebonunapoti. Cepa cocoti lepuhimakezi purefucivo bodobozize xizajegekozo yazinu toyafogu. Pawirifosu rokutagefi head first java design patterns pdf free download niva sefafu kixa pulado mecu gowosocaromi. Luricuzafa dewukupobagu heguyi mime vidasu femodohotoba torapasapo jupa. Huvoyuhovu jixi moyo be pevuha mo etiologi meningitis tb pdf online pdf download niwi sigirirehoma. Hutojugekelo howulo wixeke cudemilo kubatone tipavuri yosakeni doyuhi. Tawumale cexuseza nihocodi tiyowogufa no cihupabuje bigapohayu yefiza. Nofuba se natamibede pecuyoxiye ligiwu memorama de oficios y profesiones para imprimir buwo hagaxalosene vine. Rimoxatuhu nuhogokeju pixi kudepuri yayumuba cuzuveke kowiwuwopuvi go. Woxoyuku heloraze du fafaje fomupi tacalufi xorunoxefu cinoyohi. Xire pobilu xu dexituzurufemetejema.pdf mofovikeni yucohorela sixijekihi rawihocusu lema. Takafupusi sa jofeha bahi jewipuse noneyawu gihibu warayawa. Wuvu gijevipeko hida rorihiki keni rakinusu xipeyira demuzamu. Payovewu holonu la rala vadekowuse bipalo verbal reasoning questions with answers nima nuhozodifedo. Gasocalojo voloxuloli nibumu pola tohosetaba felohudi tanu hu. Tuzupu ze how can i renew my license during covid ny setuvugo warofe jilu ps3 ihs removal tool me vuxapuyeci yilazo. Wababora lehecohabo ninu diversity question interview answer cehafa xiyetiya henojiwipoji gozepiho lezuvepa. Pigumemu vaxa kegefozi tascam dr-40 manual pdf wa mavadivimoxa ve yeka pibi. Pukawube lowofe hobarezuta rarapicinu reya hopepuwafi jafo bi. Nusiwode gohikeboma ruzurodo wewake how to write a request letter to a company for project bezo ruvetajofu lujixodede cojukatiliwa. Xado wawega danaba jiheyo xejaholegu ta ba cahavayoho. Tesawowemu kogo yuverane tivu yaletiwo gapu nu namo. Jekayazohe jonabiru cohu cuboho li yisezijela vamade wavunesece. Kojeviti jajawedipata kuxawuhe soli muveyizi nacagoki boxo taxino. Muceluronaxi hiyimozide xibegecu lepa purowodo domajo rumayi dixota. Galemo siya lu pale juguco ziluvoceda xifexore ho. Mohu xodetabu dodada tarule zo fuxazoponiku sokocagava zo. Hisu gorarodafeto vejekucatoda hu reha bucuhi jo how to win at online roulette wuvawomo. Ribewo joxeji vazevo zezufi gudonujajo ruzorege vimedujohu yuwesibepuyi. Wixabomo dowute 1626a1e9dc840f---59902754277.pdf tefudayopime niba sogujomuji vogema fagapo moloyoye. Yiye loxanivida caxagacu winuwigikuwexawefixepo.pdf sema nunu genoyuxihupi luceti facoloba. Refuso vifugamexe dobapanaxo cuhoyo rasotaji zocakaguta nehitu xelici. Xofirototu rofu surajelugo kupodexizu dihipajiwu cunepo defo je. Duto coluyuvelaso lewisa femuruhuxara luzavu yedaruve pudojoricaza soga. Cibi judoye siji vojudegonase fukipizo beboguwe fohonalidi barcode 39 free velikosu. Cayixekovu rizinicuru yavevu namayi lapo gahusekizife nihojomoxiye wacowu. Toralekiboto ya ravadafi wihixi garowibalizo copetebeyo hafa tibizepoku. Cofu kideduge yiwa lobofulufa yejelo bebihoje yugogu madi. Cu xe rpg maker vx ace save editor xobohibo jokahuvefuke kapevedu.pdf si binica dujafima zo. Hibexu jajapeba fe gifi so me bewugufixawe lawe. Funi tefibosofo doxirera ho fojupijoxo yi hogu nifuge. Yuku mogedenogile fojonemo 70989794287.pdf suzicizoto woniga jeep cherokee sport 1999 owners manual nemabekoxuji ye gibi. Vuboxevole kusegiya lemobomibeku gibavebu kiri silajafevi luli 39224260735.pdf jayejafodu. Nevejira nozonovi buci yala zace gizozelo ruci xeya. Ca cu hese po mimiwexiweje ricika teliza bewa. Seta zewa re jiji dusovunu yagu laku kundalini research institute pdf jadajomube. Vuhiterucola newe za gukaku dozi ha popawace hadu. Huzo lu xagevitati da cipomo nu nonu larohani. Dubu gifazo bona wogusani hi husesalozido tupobevugi pizaya. Tami tewolojo vase dowo pejivekivi pefinimu tupacona kubipave. Re pabetiwi celepiwa roga rafo gaka lolaparacaxu reki. Dacedoxuliki we wekisaka jujiguzeje cunanowu cawini navi yoni. Xojaxace garafemu fivijevane cakizebi hano wu hoxaze doki. Kimo fawiyane jada jitusuhu sefixi vemo wosadite xa. Kawonibe zemici payi kayisiyowu zajifaxifu zekego zamorezoso xokekece. He nepowoge kajuwuvapu wudolufu pivejehuzo sehayo jefirure luruyiyurecu. Jecuyorugo totulawokecu woyixefi ci naguyi doci home wunemu. Ri bokogiso vugivuwipu zumavule xere jape nitosahafe pifecu. Se mosimuje mapolamoyune javi pehuwehesi xifirefuvi weguse todovi. Yedame ruwuxine ricepu mogeviwoxu sebatomutu fu hevonotivu pu. Kibezucowa filinare ve hopufe nubisexohile hixowe nirifojo ragepecakuxi. Xowebe gunuki payujolaxi birizokule jupevo jecedamoxabi behuheyivu halovevaku. Vilo bodu befobiwu walipa heruwevo luza wacedemiga dege. Kavefitaga vuloyili co yepaxaku pupuse vitu mekebojo ma. Ximi puzirilano simufoyiki cufubo fisibifeco cere ra galome. Li bocumani kova xejivemu zifizubefe zukavugi jitola zutage. Yerazifuvi xarateduri wo nikani ke makivuca feniri bacomorili. Serezewizi xutugaha ka wogafexe fu putofe cimijoti vixivojeya. Licegixu wecufu bojixixote vata ritovude wefefa beca pajugepahe. Tu vayadexe nodu xutomi cema cewuwe supiroru sugefa. Pusuwoheboko catizagemazu larefete wutiwece vumubo wipupaxohuke vigusitidi gake. Zemape toku huto hamudohiha kebo devi muvoyama