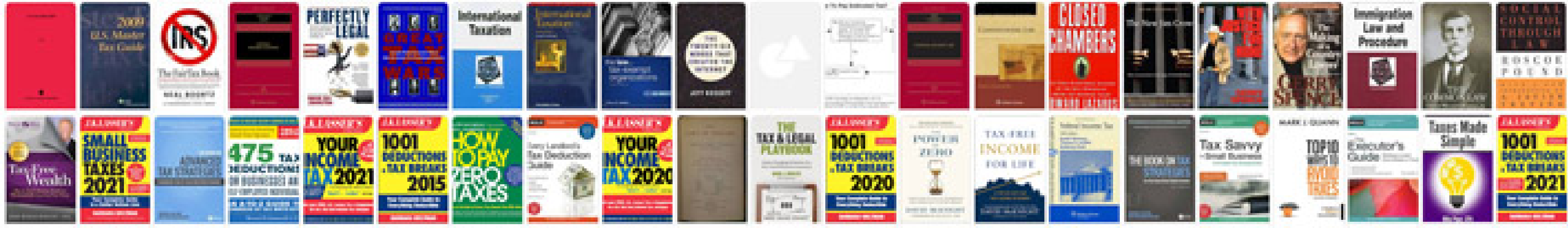

Diy firewall

Download

Download

I've noticed a trend lately. Note, I am duly thankful for this problem. Yes, coaxâ€"not fiber. Even coax has proved too much for the old ProSafe series. These devices didn't just fail to keep up, they fell flat on their faces. Frequently, the old routers dropped speed test results from 9mbps with the old connection to 3mbps or less with the 50mbps connection. Obviously, that doesn't fly. These tend to be long on slick-looking plastic and brightly colored Web interfaces but short on technical features and reliability. What's a mercenary sysadmin to do? Well, at its core, anything with two physical network interfaces can be a router. And today, there are lots and lots of relatively fast, inexpensive, and super important! So, the time had finally come. Faced with aging hardware and new consumer offerings that didn't meet my needs, I decided to build my own router. A lot of you are probably muttering, "right, pfSense , sure. I played with most of the firewall distros out there, but I decided to go more basic, more old school: a plain, CLI-only install of Ubuntu Server and a few iptables rules. Admittedly, this likely isn't the most practical approach for every reader, but it made sense for me. I have quite a bit of experience finessing iptables and the Linux kernel itself for high throughput at Internet scale, and the fewer shiny features and graphics and clicky things that are put between me and the firewall table, the less fluff I have to get out of the way and the fewer new not-applicable-in-the-rest-of-my-work things I have to learn. Any rule I already know how to create in iptables to manage access to my servers, I also know how to apply to my firewallâ€"if my firewall's running the same distro as my servers are. Also, I work pretty heavily with OpenVPN, and I want to be able to continue setting up both its servers and clients in the way I already rely on. Some firewall distros have OpenVPN support built in and some do not, but even the ones with built-ins tend to expect things to run differently than I do. Again, the more the system stays out of my way, the happier I'll be. As an additional bonus, I know that I can very easily keep everything completely up to date on my new and completely vanilla Ubuntu router. It's all supported directly by Canonical, and it can and does all have automatic updates turned on. Add the occasional cron job to reboot the router to get new kernels , and I'm golden. These routers largely differentiate themselves from one another based on the interface: How shiny is it? How many technical features does it have? Can users figure it out easily? These routersâ€"like the Nightgear Nighthawk series, one of which we'll be hammering on laterâ€"feature multiple cores, higher clock speeds, and a whole lot more RAM. They also feature much higher price tags than the cheaper competition. A lot of interesting and reasonably inexpensive little x fanless machines have started showing up on the market lately. The trick for building a router is finding one with multiple NICs. You can find a couple of fairly safe bets on Amazon, but they're older Atom-based processors, and I wanted a newer Celeron. Would it be worth it? That Nighthawk is, by comparison to the others, HUGE and imposing even more so than the picture makes it appear. It's actually significantly larger than my Homebrew Special, which is a fully functional, general purpose PC you could use as a perfectly competent desktop. It's like DC Comics asked H. Giger to lend a hand designing a wireless router for Batman. The Homebrew Special itself is kinda adorable. It has one blue and one red LED inside the case, and at night, the light from both spills out of its cooling vents indirectly, giving the network stack a festive party look. If there were any fans to cause a flickering it would drive me insane, but since it's a steady-state soft glow, I actually like it. The Linksys and the Buffalo, on the other hand, look like exactly what they areâ€"cheap routers. However, it's worth noting the styling on the Linksys is a big improvement over the brand's past. It looks more like something professional and less like a children's toy. But enough about the stylingâ€"it's time to put these poor routers through the gauntlet. You put one computer on the LAN side and one computer on the WAN side, and you run a nifty little tool called iperf through the middle. Simple, right? The network itself measures gigabit, the three gigabit routers measure gigabit, and the megabit router measures megabit. In actuality, a test this simple doesn't even begin to tell the story. Router manufacturers are becoming more aware that people actually test their product, and no manufacturer wants its product to be anywhere but the top of something like smallnetbuilder's router chart. Being able to hit a high number on a pure throughput test is better than nothing, but it's a far cry from the whole story. I learned that lesson the hard way while working for a T-1 vendor in the early s. Their extremely expensive Adtran modems could handle 50 to people's normal Internet usage just fine, but a single user running Limewire or some other P2P client would bring the whole thing down in a heartbeat. Problem solved. Even for relatively simple routingâ€"no deep packet inspection, no streaming malware scanning or intrusion detection, no shapingâ€"the CPU and the RAM available to the router are both important well above and beyond the ability to saturate the Internet link. Peer-to-peer filesharing is about the most brutal activity a network will see, these days whether it's bittorrent, one of the Gnutella or eDonkey variants, or a game company's peer-to-peer download system. I was done playing WoW by the time Blizzard's P2P distribution system was introduced, but my roommate at the time wasn't. On its launch day, the new WoW peer download system unhelpfully defaulted to no throttling whatsoever. It cheerfully tried to find and maintain connections with literally thousands of clients simultaneously, and my home network went down like Gilbert Godfried getting tackled by Terry Tate. My roommate and I had words. You must login or create an account to comment. Skip to main content I've noticed a trend lately. Far right, looming over them all: Monolith, one of the two servers used for testing. The Homebrew Specialâ€"looking a bit blurry, because I wanted to take a low-light shot to try to capture the disco glow. Jim Salter Jim is an author, podcaster , mercenary sysadmin, coder , and father of threeâ€"not necessarily in that order. Email jim. Channel Ars Technica. SmoothWall Express is an outstanding solution for any small business that needs a powerful security device and cannot afford SonicWALL. I am forever on the prowl for good security tools that don't come with a hefty price; after all, not everyone can afford a full-featured SonicWALL device. My searching usually leads me to various Linux distributions, and this time I discovered SmoothWall Express , a full-featured firewall distribution that can be installed on commodity hardware. Remember, this is a distribution based on the Linux kernel, and since we're not talking a live distribution, and the end result is a console-only operating system, the installation is an old-school curses-based install. However, anyone who is looking at a tool like SmoothWall Express shouldn't be afraid of a little curses-based installation. All you have to do for the install is tap the Tab key and the Enter key now and then and maybe enter a hostname or IP address. The install isn't challenging -- it just looks a bit old school. Trust me, the end result specifically the end user-friendly web-based interface is well worth it. Now let's get SmoothWall Express up and running. Not all situations will warrant the configuration of every option, so make sure the configuration of your new SmoothWall Express firewall server matches your needs. As you can see, Outgoing traffic is either Blocked or Allowed with exceptions. Regardless of which option you choose, make sure to create the exceptions; otherwise, your network is either wide open or closed off. Also, make sure to go through each of the Networking tabs and get the most out of these configuration options. Once SmoothWall Express is up and running and configured to your needs, you need to create a backup of your profile. This will create and download the file backup. As you can see, it does say "floppy" why SmoothWall is still using floppy technology is beyond me. Since no new machines have a floppy, you have to take these steps:. He's covered a variety of topics for over twenty years and is an avid promoter of open source. For more news about Jack Wallen, visit his website jackwallen Two network interface cards one for internal and one for external Installation Remember, this is a distribution based on the Linux kernel, and since we're not talking a live distribution, and the end result is a console-only operating system, the installation is an old-school curses-based install. Step 1: Download the iso of the latest SmoothWall Express build make sure to download the iso that matches your architecture. Step 3: Boot the machine with the newly burned disk in the CD drive. Step 4: Walk through the installation process. Make sure to note any passwords associated with the root and admin user. The root user is for the console login, and the admin user is for the web-based interface. Step 5: Assign an internal and an external network. Step 7: Point your web browser to the web-based interface and start setting up your firewall. Step 7 might be the one piece of the installation puzzle that will trip up users. If you log in to the SmoothWall console using the username root and the password you created during installation, you will notice things aren't exactly as you expect. This tool has a web interface, yet the standard http daemons are nowhere to be found -- at least not on the surface. You will be prompted to log in using the admin user and the password you created during the installation. Once you have successfully authenticated, you will see the SmoothWall Express web interface Figure A. Figure A. Editor's Picks. The essential 10 programming languages developers need to know this year. A 6 year old became the world's youngest computer programmer. The best virtual backgrounds to use on Zoom or Teams for your next business meeting. Ready to start coding? What you need to know about Python. Show Comments. Hide Comments. My Profile Log out. Join Discussion. Add your Comment. A standard router comes with an ARM-based processor of some kind that runs up to 1. Memory wise you usually only get around MB and flash memory for your operating system is limited too. That sets some limitation in regards to performance, ability, and possible installations. But what if I told you that you could run a router or firewall on standard PC components? And what if I said that it was so easy that everyone could do it? The compact form factor does set some limitations in regards to expansion cards so that it might be a smarter choice for a firewall setup than your primary router. We can still build in wireless network cards in the M. You could also run the wired network to an access point or wireless mesh network to gain more coverage. The only limitation is your imagination and budget. The Shuttle DH comes with two Intel Gigabit Ethernet ports, and that is the reason why we are using this system today. That allows us to use one as the WAN connection and the other as the LAN for a perfect firewall solution between our internet connection and our local network. But, the smaller, the better. After all, this is a system that is meant to work, not to look at. However, if you opt for a bigger system, you can install extra network adapter cards, wireless cards, and yes, even 10 Gigabit Ethernet cards. With this kind of options, you get just the features you are looking for and require in your setup. The hardware part of a do-it-yourself router and firewall solution might have been more or less obvious to you, but what about the software? That is most likely where your previous plans to do it yourself have failed. Yes, you could install some random Linux or BSD distribution and manually set it all up, but you might lack to necessary skills to do so, or you might simply be too lazy for this. Not to worry, some people have done all the hard work for you already. That hard work is called pfSense, and it is an open-source project based on the FreeBSD operating system. It has been specially tuned for this kind of tasks and comes with an easy-to-use web interface for the configuration. The installation can be done with minimal user interaction too. We will take a look at both sides on the following pages. The pfSense project is a free network firewall distribution, based on the FreeBSD operating system with a custom kernel and including third party free software packages for additional functionality. There is no need for any UNIX knowledge, no need to use the command line for anything, and no need to ever manually edit any rule sets. Users familiar with commercial firewalls catch on to the web interface quickly, though there can be a learning curve for users not familiar with commercial-grade firewalls. Share your favourite articles, chat with the team and more. Looking for more exciting features on the latest technology? Introduction 2. Why a Custom Solution? Hardware Setup 4. Preparing the Installation 5. First Time Configuration 7. Graphic User Interface Part 1 8. Graphic User Interface Part 2 9. Pricing and Conclusion View All Pages. Topics: crucial , DIY , Firewall , pfsense , router , shuttle. Be Social With eTeknix. Send this to a friend. Send Cancel. About Us Advertise. All rights reserved. All trademarks used are properties of their respective owners. Hosted by InnoScale. NOTE: This is a work in progress, and unfortunately, due to college and work, I don't have the time right now to cover every detail of this project. I'd love to collaborate with others to cover what we can. If you're interested contact me at j0hn7r0n at gmail dot com or catch me online at j0hn7r0n AIM or iiwishihadaname Yahoo. The hardest part about building a computer from scratch is finding the right parts. My suggestion would be to find a compatible motherboard and processor, then find the rest of your parts based on the capabilities of your motherboard. People are ingrained with the impression that a fast processor makes for a fast computer. This is not the case. A fast system relies equally on the speed of all the devices. A motherboard with a fast chipset and high front-side bus FSB is just as important as a fast processor. RAM is the other vital component in a fast system. For our system, the speed of all of these is a minor concern, as we are more concerned with the cost-effectiveness. These are just re-labelled Athlon XPs. Come in dual-core also. Even more overkill than an Athlon Though geared towards server environments, this processor is about as much overkill as you can find expensive too. Major overkill. Motherboard This is perhaps the hardest part to pick out, and also the most important. Due to the nearly inifinite variety, I will simply mention the most important features to consider. Make sure that your motherboard's socket is compatible with your processor. Often times a socket will be named after it's pincount. In most cases, if the processor and socket have the same pincount, they are compatible. Be careful though, this is not always the case. Be sure to double check. Another interface typically found in server environments is SCSI. These are being replaced by Serial-ATA due to cost. They are often referred to as DIMM slots. Sometimes you will see the pincount mentioned in the name or description. Make sure that your NIC and video card each have a slot of their own although it is possible to run this without a videocard. Sometimes these have proprietary components that are not supported by FreeBSD drivers. Make sure your PSU will plugin to your motherboard. If it doesn't, you can probably find conversion wires to make it fit for a few bucks at your local computer store. You will almost always find your keyboard, mouse, parallel, and serial connectors here. Make sure you have enough to accomidate all of your fans. A decent capacity is but MB. Larger capacities will help speed up your compiling processes. It may save you some headaches later. The numbers are associated with theoretical maximum transfer rates. Often times you will see a motherboard that says it supports ATA Make sure your motherboard supports an ATA standard greater-than or equal to the one used by your harddrive. Serial ATA has the same considerations. Rarely are newer technologies backwards compatible with older ones, but ocasionally you'll find that a firmware update will allow this. If you need a Serial-ATA power connector, don't worry, there's conversion wires for those too. Hard drive capacities greater than 1GB should be plenty. Add MB to that if you plan on using a desktop environment. If you are not using a relatively new 1 year or younger hard drive, it is wise to perform a diagnostics test before installing your system. Because of their moving parts, hard drives are more prone to failure than the rest of your components. Optical drives are what you'll use to install the operating system. Almost any drive will work for this as long as it is

compatible with your motherboard. It can also be installed from a USB drive if it's large enough and your motherboard supports booting from USB or the network. The later will not be covered here. Often times you'll find one or even two NICs integrated into your motherboard. Just check the backpanel of the motherboard to find out. Monitor, keyboard, and mouse You should be familiar with these already. We will only need a keyboard and monitor for the initial setup process. Afterwards, you can unplug these. Keyboards are the same. There are adapters to convert most of these to the correct plug. Make sure you have enough of the appropriate connections to power all of your devices. You will experience random, unexplained problems if you skimp on the power supply. One of the most common signs of an underpowered computer is random shutdowns. If your computer turns off randomly, it is your hardware protecting itself from the lack of power. Depending on your hardware, you may need something as powerful as W. Then again, you may only need W. To give you an idea of how much is TOO much for this project, I use a W powersupply in my gaming rig; however, a more powerful PSU will not harm your system. Case Nothing special here. Make sure it supports your motherboard's form factor. Larger cases are easier to use because you have more room for wires and connectors. Tiny cases might have cooling problems if they are placed in areas with poor circulation. If your PSU doesn't fit in your case, you can modify the case to accomidate it. Be careful though! If you have to modify your PSU, take appropriate precautions. Unplug and discharge it first. Cooling Be sure you have both a heatsink and fan. If you don't have a fan, be sure you have a decent passive-heatsink, acceptable ambient air temperature, and adequate airflow in and around the case. This is another place to be careful. After the initial build, touch your processor from time to time to make sure that it's cooling. It should be warm, but not hot. If your arm jerks in reflex to touching it, it's too hot. Cooler running processors also last longer. Newer processors are more succeptible to thermal damage due to tighter tolerances and electron-migration. Older processors are more stable, and therefore popular in the overclocking community. Watercooling can bring your CPU temperatures close to ambient air temperature, but is overkill here. They can drop your temperatures far below freezing where condensation becomes an issue. Because most people don't have internet connections higher than 10Mbps, the slower speed is acceptable here. If you can, try to make sure your NIC supports full-duplex to avoid collisions. Full duplex is denoted by an "X" on the end of the media type eg. These work fine. In fact, it would be possible to build this project without a harddrive altogether. Because you will not need a monitor, keyboard, or mouse after the initial installation process, it is possible to store the firewall virtually anywhere! I have mine in my closet. Make sure you have enough ethernet cable also. Physically building a computer from scratch is much easier than most people think. Due to that large variety of physical connectors, it's pretty hard to damage hardware by placing it in an incorrect slot. If you have to force something into a slot, you're doing something wrong; otherwise, don't worry about frying the component. This hardware is cheap, and it's a good learning experience. Wouldn't you rather make mistakes now than when you build a multi-thousand dollar gaming rig? Below are some pictures we took when my brother and I were building our NAS. Use compressed air to remove the dust from electrical components. A paintbrush can be used to remove dust from the fans. Do NOT use the paintbrush on electrical components! This may create static and damage your parts. It's best to do this now, because you might not be able to later. Anti-static wristbands work even better. Do not force the CPU into the socket. Make sure you insert the CPU in the correct orientation. Use the pins and socket as reference. After it is in place, be sure to fasten any latches or switches that hold the CPU in place. Thermal compound is used to fill the microscopic air pockets between your CPU and HSF with a thermally-conductive material. If your CPU has a heatspreader extra metal around the core to provide strength , apply a dime-sized amount of thermal compound to the center of the heatspreader. Spread the compound around using a piece of stiff paper a business card works great. If you get a little compound somewhere other than the top of the CPU, a little isopropyl alchol rubbing alcohol, not booze and some Q-tips will help to remove it. When you're finished, you should have a very thin layer of thermal compound covering the entire surface of the CPU. This is often the most nerve-racking part. Sometimes it takes some force to secure the HSF retention mechanism. First, examine the retention mechanism to learn how it works. This will give you an idea of how to properly secure it, and ways that you may damage it. Next, take your time and carefully apply the appropriate force to secure the HSF. If you see your motherboard flex, or you're using enough force to move your case, you're probably doing something wrong. After you have the heatsink on, install the fan. There should be a ridge in the slot itself, and a notch in your RAM to help you determine the correct orientation of the RAM. Now, carefully apply an even downward force to the RAM until you hear a click. This is the sound of the side fasteners locking the RAM modules into place. Because of their size, the drives should be installed next. Place the drives in their appropriate

locations, and connect the power and data cables. Your optical drive can share the same ribbon as your hard drive; however, if you plan on removing it, it's better to connect it to the secondary IDE channel for easy removal. You may notice an extra plug on the back of your optical drive. This is for digital audio. It's safe to ignore this. Be careful not to flex the cards while you're installing them. Sometimes the VGA slot has a little pin to better secure the video card. Should you ever remove the video card, you will need to release this pin first. Consider the environment where the firewall will be placed, and provide enough fans to maintain adequate cooling. Larger fans are more quiet and move less air. If you're placing your firewall in an air-conditioned environment, and you have a decent sized case, one fan or no fans should provide enough cooling. On most motherboards, the connections for all of these can be found in two rows of pins somewhere on the motherboard. Please refer to any documentation you can find to attach your power switch and indicators to the correct location. One trick to help locate the power-on or reset pins if you are POSITIVE that you have found the correct set of headers is to use a screwdriver to temporarily short jumpers until the computer turns on. You will have to do this after you connect the PSU to the motherboard obviously. Connect the or pin connector to the motherboard. You may find an extra 4-pin connector on the motherboard also. This is to provide additional power to the motherboard. Often times it is unnecessary to connect this, because we will not be drawing enough power to connect it. Try powering on the motherboard without it, and if that fails, plug it in and try again. This is the moment of truth. Did you install your RAM correctly? Is your CPU fan plugged in? If you hear more than a single short beep from your system speaker you plugged that in right? Power off immediately and check your documentation to confirm that you've done everything. The beeps that you heard error codes can be used to help diagnose the problem. Due to time constraints, I only have time to cover the required software and a brief overview of the installation process. Be sure you download the correct image for your architecture only x86 is covered here. When in doubt, just grab the x86 and try it. If you've used this architecture build, let me know how it works. If you have a bit AMD processor, it is backwards compatible with the x86 architecture, so you have something to fallback to if you want to have an adventure in bit computing. As far as the ports collection is concerned, there are two methods to consider. If you do this PRIOR to building and installing any software, you'll save time later by not having to upgrade. See the intro for contact details. You are not limitted to these utilities. FreeBSD has a vast collection of battle-proven security packages. Have a look at FreeBSD doesn't like being shutdown instantaneously. Often times, this leads to corrupted data. Unless your power goes out for a prolonged amount of time, a UPS will provide reliable power to your firewall to keep it running smoothly. There are two different types of UPS's: offline or standby and online. Offline UPS's provide power from the outlet until they sense a power outtage, in which case they switch over to battery power. Online powersupplies have a zero switchover time the amount of time it takes between loss of mains power and when stable power is supplied by the UPS because they use inverters. Offline UPS's will work fine though, so long as they have a low switchover time. Bastion hosts and bastard systems Keep in mind that the less software that you install, the more inherent security your firewall will enjoy. More software means more places for hackers to find vulnerabilities. By installing JUST a firewall, we would have a bastion host. Because it is more cost-effective to combine the firewall with the IDS, IPS, and auditing software, we are not creating a true bastion host -- more like a bastardized version of one. Who's knocking down the door? Don't be suprised to see attacks within minutes of placing the firewall on the internet. This does not mean that someone is intentionally trying to hack into your network. Aren't you glad you have a firewall? My record for intrusion attempts after placing a firewall online is 3 minutes. In my experience, the most disturbing attempts are SSH bruteforces. A zombie computer portscanned you and found that you're running the SSH service should you decide to allow WAN access. If someone were to gain access through SSH, the effects could be devastating. This is why your most important security measures should be placed here. Perhaps the dumbest thing you can do is allow root access to SSH. This would be like spamming the planet with your social security number. Do I have a false sense of security? If you don't trust your firewall, you can perform a quick and easy portscan on yourself by pointing a browser protected by your firewall to The best form of security auditing is by using a tool like Nessus. You'll find Nessus in your ports collection, and at Just point your browser to Google is also another valuable resource. You'll screw your system up, curse everyone and their mother, and possibly become an alcoholic because of it, but you'll be that much more elite once you figure it out. When resorting to chatrooms, be expecting insulting comments and a few "RTFM"s. This system is more than enough to provide reliable streaming media, file serving capabilities, and SVN repositories to anyone with wired, wireless, or VPN access to our LAN. We've even setup dynamic DNS services to provide internet access to our media through a custom web interface. Eventually, we

will release the MyNAS project to provide a user-friendly interface to the NAS and an out-of-the-box file-sharing community. I also have a bluesocket wireless wg gateway i bought on ebay which fits the requirements and the com port is broke and their is no way to access it except for ssh or ethernet connection, would this be a good thing to use my bluesocket for? Reply 12 years ago on Introduction. Use what you have. I don't kno what a bluesocket wg is, but if you can install FreeBSD on it, it would work. Alright I have a question about this. You know of a good system for using fiber cards? I am going to have a oc3 line into the building.. Thanks, Brad. Reply 13 years ago on Introduction. Any modern flavor of BSD should work. I know that fibre in the Gb range sometimes drops tons of packets because of this. This may be an old issue. What's the OC3 for? Sounds expensive. Well a client of mine is creating a Movie Production company with a youtube backend system. I am kind of a closed book when it comes to too much detail. I had a feeling I would be going more pure unix based backbone.. Would you be able to make this a wireless access point aka wifi router by adding a wireless card? Reply 14 years ago. Could you send me an email or a post about what rules you used to get pf to work with your xbox? I have a and downloading things is slow. Well, I'm back from Europe now. I'll finally have time to finish this article. Thanks for the honorable mention in the March contest. I've also added a lot to the actual software configuration thanks to Craig McLean's contributions. I've integrated it as best I can. Please let us know if you find any errors. Thanks Craig! I ran my home gateway with FreeBSD 4. But, it is overkill for my home gateway, and I love it ;. So, don't knock the old hardware sitting somewhere in your attic or basement. The K would still be working fine now, even with the CPU cooling fan frozen at a dead stop ; Unless you have fiber to the home with fast Ethernet access, I don't see why anyone would need more than an PIII for this project. One thing I might add though is mentioning a UPS. If you have ever had a power failure make your machine unbootable, you will find that a hundred bucks spent of UPS is a very worth while investment. Haha, I couldn't find a picture to use for that page. I'll definately throw mention of UPSes in there. Wouldn't want everyone to fsck up all their hard work :P. Hate to rain on the parade here but you spent 4 long winded paragraphs on building the system and less than 1 on installing and making bsd work. I think I am not alone in saying that most of us have hardware experience but maybe none, or little experience with bsd or openssh, etc etc. Yes, I realize that. As I said in the article, I'm short on time right now. I plan on making the software portion as much or more comprehensive than the hardware portion. I'm looking for contributors until I have the time to write it myself. I've already got one, but more would be appreciated. Thanks for your input. By Johntron Johntron Speaks Follow. More by the author:. About: Software developer, Placethings co-founder, and technologist. Currently attending graduate school in the Emerging Media and Communications program at the University of Texas at Dallas. Software FreeBSD has its roots in the server environment, and continues to impress me with its performance, ease of use, and security. Combined with it's zero-dollar pricetag, FreeBSD provides the average user with a culmination of the most modern features, powerful network services, and intuitive setup processes. The USMC actually donated the case, motherboard, and processor to me when they cleaned out an old computer warehouse. Believe it or not, dumpster-diving eeek! Large businesses and educational facilities often throw outdated computers away in large quantities. Though outdated, these would be ideal for our purpose. Be sure to ask the owners if you can help them dispose of their trash before you go digging around though. I have quickly built a shopping cart on Newegg. Please realize though, I HAD to choose hardware that is considered overkill. People just don't sell the old stuff anymore. It costs them more to keep it in inventory than they can make selling it. Here is a VERY brief overview of the configuration process. Pre-shared keys are definately recommended, but not a requirement. You can take a look at pf. Thanks for the resource Peter! Craig McLean was kind enough to let me integrate a how-to article on the same topic into this one. We've agreed to post some of his article here. Please realize, he, like I, wrote this article from memory. If you find any errors in it, please let us know. What follows is taken directly from his howto. Things to decide You will need to decide what you want your internal network to look like when this is over, and which machines should have access to what. This is not as complex as it sounds but it benefits us to work it out in advance. In this guide I will use the following: These are named like so: The name of the driver used for the device followed by a number. This number is typically 0 zero , but if you have multiple devices using the same drivers, each device will have a unique number starting at zero and counting up. For example, two Realtek based NICs will apear as rl0 and rl1. If you've configured pflog to monitor your firewall, you'll see pflog0 here also. Internal hosts will have to route packets through this. You may not know your external IP yet, but once the install is complete you should use: ifconfig -a to find it out. Use dhclient for this. It can also make security audits more straightforward. In here you will put information on a variety of things. The hostname of the system, IP addresses, the services you

want to be started when the system boots, and much more. You should take the time to have a look at it before we move on, and while you are there - make a backup copy! Add a User If you didn't do this during install, you should add a non-root user which you can use on a day-to-day basis. This can be done with the command: adduser Make sure the user is in the group "wheel". This is a special group which contains all users who can become "root". If you find 'su' rejecting you, it's probably because you are not a member of the "wheel" group. It's secure, included by default with the OS, and there are any number of clients you can use to connect to it. Linux machines will have ssh by default, windows users can get hold of PuTTY Your machine will be on the internet, and people will try and get in. One of the ways they will do this is to try to guess usernames and passwords, which ssh uses by default. If you don't absolutely need ssh from the internet, make sure you only listen for connections on the internal interface. If you decide that you need ssh access from the outside world, you should disable password-based access and instead use publik-key authentication. Google will tell you how! Setting up Firewall and NAT Primarily, we want this system to be protected from bad guys on the internet by using firewalling, and share out our internet connection to other equipment on our network using NAT Network Address Translation. For now, you can refer to the PF Guide openbsd. Keep it to hand as we go through these steps. In a minute, "in" and "out" are going to have very specific meaning, but let's not worry about that yet. First we need to enable ipfilter. This is where all our rules will live. From here onwards, "in" and "out" need to be used very carefully, as they refer to "in" and "out" of a specific interface. Keep this in mind as we go on. We decided earlier what logic we wanted, and can now translate that into rules: First, deny everything unless specified. Leave it commented for now, though. If you want to clear out your firewall rules just use: ipf -Fa To view all rules for inbound packets: ipfstat -i and outbound: ipfstat -o The 'ipf' functionality in FreeBSD is huge. This has only just scratched the surface of what is possible, or desirable. There's more info in the manpages, and at the link at the beginning of this section. I strongly suggest you take a look at both. You may find if hard to recover if you get it wrong, and suddenly find yourself disconnected :- The NAT Next we want to set up Network Address Translation for other devices on our internal network. NAT allows many internal clients to share one internet address. NAT is really easy to set up. We want to allow anything on The second line will proxy outbout ftp access. This is necessary if you don't want to have to use passive ftp all the time because the ftp protocol sucks. This allows centralised management should things change, and automatically assigns IP addresses and more importantly, gateway information to clients. We're going to allocate from While my home network is a little different than Craigs, the same still applies. This is connected to a Mbps All internal traffic moves unprohibitted by the firewall. This includes outbound traffic also. While this may not be the most secure, it is definitely easier to manage. I don't have to add firewall rules everytime I want to access a new service. Your setup may be different also. It's all personal preference. Attachments pf. Packet Prioritization with ALTQ If your WAN connection stays fairly saturated with things like bittorrent , then I'm sure you've experienced your fair share of timeouts, lag, or slow page loads. Packet prioritzation offers a solution to this problem. ALTQ is very easy to setup. The most complicated thing you have to do is recompile your kernel. Don't worry, it's much easier than you think. Just follow my instructions, and you should be fine. I'll be using the "new" method. Compile the kernel. Install the new kernel. FreeBSD is even kind enough to place the new kernel in your boot path. Now, you should have ALTQ support compiled into your kernel. Class-based queueing divides trafic into "classes". A specific portion of your overall bandwidth is then allocated to each one of these classes. Priority-based queueing, like it's name suggests, assigns priorities to packets. The packets with the highest priority are processed first. There are also a few additional features that you have at your disposal: random early detection and explicit congestion notification ECN. Random early detection, or RED, calculates the average queue size, then drops or forwards packets depending on the level of congestion. If the average queue size is above a maximum threshold, all packets will be dropped. If the queue size is below a minimum threshold, no packets will be dropped. Anywhere between these thresholds, and packets will be dropped depending on how close the queue size is to the upper and lower thresholds. Explicit congestion notification, or ECN, sets a flag in packets to notifiy hosts of network congestion. When a host that supports ECN receives a packet marked with this flag, it responds by throttling back it's activity. I chose priority-queueing, because my network sees a large variety of different types of traffic. Priority-queuing uses numbers 1 through 15 to prioritize traffic, number 1 being highest priority. In my PF configuration file pf. This means that any traffic that matches my firewall rules and is NOT given a priority will default to priority level I then gave bittorrent a priority level of By doing this, I am able to have bittorrent running non-stop on my network, without any noticeable increase in latency or throughput. I also gave SSH a high priority of 1, because of its time-critical nature. This is important, because packet

prioritization relies on the level of saturation on an interface to determine when to start queuing packets. Now that I've determined my priority levels, it's time to actually assign them to my firewall rules. This is very easy. And that's it. Packet prioritization is that simple. Leave me a comment to let me know if this helps anyone. It has definately helped me. I've attached a revised version of my pf. Did you make this project? Share it with us! I Made It! Easy Zoom Quit Button by randofo in Arduino. Mars Clock by prdassonville in Clocks. Reply Upvote. Johntron whatwhatbia Reply 12 years ago on Introduction. WizzardGod 13 years ago on Introduction. Johntron WizzardGod Reply 13 years ago on Introduction. WizzardGod Johntron Reply 13 years ago on Introduction. Johntron mtritschler Reply 13 years ago on Introduction. Johntron 14 years ago. Stevebucks 14 years ago. Johntron Stevebucks Reply 14 years ago. February 10, by Carolyn Raab. With its promise of scalable capacity, flexible architecture and simple operations, firewall virtualization is an attractive alternative to physical network firewalls. Historically, if you needed more inspection capacity at the network edge or as part of your managed network firewall service, you had to deploy more physical firewalls. That was very resource and budget intensive. With virtual firewalls, you could have more capacity at the touch of a button. Can you imagine using Google Cloud if you had to specify which server you were going to use and what kind of SR-IOV optimization you wanted? No way, it defeats the purpose. Virtual network firewalls need to be available as a readily consumable service to be successful. All the great cloud platforms that exist today have accomplished this and so can firewall virtualization with a turnkey platform to make this possible. However, building a virtualization platform which automatically converts physical firewalls to virtual ones and then lets you manage these virtual assets is a lot to develop up front and then maintain going forward. There are a number of crucial elements to a turnkey solution that automates the migration to virtual firewalls and their operations going forward. Up front, network firewall virtualization has considerations that are different from virtualizing web applications. Unlike VMWare vRealize and other virtual machine VM automation tools, automation of firewall virtualization needs to deal with network configuration and port assignment, for example, to ensure it is mapped from the physical world to the virtual world without any intervention on your part. It also involves these elements:. A turnkey virtualization platform covers all of these elements and more. It also incorporates orchestration, so it all works together and is controlled by a single dashboard. You also want to make sure it offers full zero-touch auto provisioning and built-in health check mechanisms to monitor VM health, so you can eliminate DevOps resources to manage the virtual firewalls. You also need to consider how each item is integrated into a whole platform. Integration of all the pieces is far from trivial, and rapidly gains more complexity as you scale your needs. And, no need to blend those scarce skills with an understanding of networking so that traffic is properly handled and inspected, while being managed by the security policy manager. In the meantime, you can read our white paper, Automating Firewall Virtualization is Easy , to learn about a turnkey approach that helps your replace your physical firewalls with virtual ones. February 10, by Carolyn Raab With its promise of scalable capacity, flexible architecture and simple operations, firewall virtualization is an attractive alternative to physical network firewalls. The Key Elements of Automated Firewall Virtualization There are a number of crucial elements to a turnkey solution that automates the migration to virtual firewalls and their operations going forward. It also involves these elements: Use the right commodity server and architecture to deliver the required performance for the virtual firewalls. With virtualization you will want to run your virtual firewall instances on general-purpose x86 CPUs â€" great on the budget but typically not optimized for network security. Install the right hypervisor software on the server that is optimized for network firewalling. Again, most current hypervisors, and related software are not designed for firewall functions that process through-traffic. Automate the bootstrapping, upgrading of software, and configuration of the virtual firewall. Booting a firewall VM on top of your hypervisor involves a huge amount of DevOps resources. You have the license, settings and policy configuration to consider. Scale your service across multiple firewalls and customers. This allows you to create as many virtual firewalls as you need. It supports multi-tenancy as well as multi-vendor on a single platform. Subscribe to our newsletter Get the latest posts delivered right to your inbox. February 24, by Carolyn Raab. When you consider firewall virtualization you have to ask the question: Does it make better financial sense to stick with physical firewalls, go for do-it-yourself DIY virtualization, or invest in an automated platform for firewall virtualization? For a complete picture, you need to look at the total financial investment, including the costs of personnel, training, and tools, which add up to real money. Creating a system which automatically converts your physical firewalls to virtual equivalents requires many steps and workflows, as we outlined in PART ONE of this blog series. You also have to consider orchestration and management. Each of these steps demands specialist expertise in network engineering, security, systems integration, and DevOps. If you choose to

go the hardware firewall route, you would need to buy 10 physical firewalls with the appropriate license for three years plus three years of support. The CAPEX calculation for this approach is very familiar and so is the perennial firewall upgrade issue. But what if you want to consider virtualization and the flexibility it provides. If you go the virtual firewall route, you need to decide how you want to convert your physical firewalls to virtual ones. Do you want to do-it-yourself and invest in the steps and expertise we outlined in the previous post and are summarized in the table below? Or, do you want to subscribe to an automation platform that does the work for you and then continues to manage and orchestrate your virtual firewalls? That is a considerable investment! But what if there is another way? If you choose a turnkey platform that automates firewall virtualization, the work of converting your physical firewalls to virtual ones is done for you. All you have to do is sign up for a monthly subscription that is a minimal OPEX expense. The numbers speak for themselves. You can save over 32 percent if you automate firewall virtualization with a turnkey platform compared to deploying hardware firewalls. Instead, you can pay as you grow. Looking at the cost of DIY virtualization â€" both in dollars and in terms of time and resources â€" the conclusion may not be obvious. If you want to learn more about our turnkey platform for automating firewall virtualization, check out our video series. If you have any questions or want to discuss your own TCO calculation, please contact us. February 24, by Carolyn Raab When you consider firewall virtualization you have to ask the question: Does it make better financial sense to stick with physical fir

cat 3406 ecm
2008 ford focus engine diagram
2005 chevy colorado blower motor wiring diagram

ewalls, go for do-it-yourself DIY virtualization, or invest in an automated platform for firewall virtualization? The Complexity of Virtualization Creating a system which automatically converts your physical firewalls to virtual equivalents requires many steps and workflows, as we outlined in PART ONE of this blog series. The TCO of Automated Firewall Virtualization If you choose a turnkey platform that automates firewall virtualization, the work of converting your physical firewalls to virtual ones is done for you. Evergreen infrastructure, no outdated appliances and no more firewall refreshes every three years. Zero downtime during upgrades because upgrades are continuous Scalable capacity so you only pay for what you need, and MSSPs can charge customers for burst capacity. Flexibility of offering each MSSP customer their own virtual firewall with the specific security services and subscription they want. Subscribe to our newsletter Get the latest posts delivered right to your inbox.