Securing a linux webserver for public access

Asked 16 years, 4 months ago Modified 11 years, 2 months ago

Viewed 4k times

▲

21

▼

I'd like to set up a cheap Linux box as a web server to host a variety of web technologies (PHP & Java EE come to mind, but I'd like to experiment with Ruby or Python in the future as well).

I'm fairly versed in setting up Tomcat to run on Linux for serving up Java EE applications, but I'd like to be able to open this server up, even just so I can create some tools I can use while I am working in the office. All the experience I've had with configuring Java EE sites has all been for intranet applications where we were told not to focus on securing the pages for external users.

What is your advice on setting up a personal Linux web server in a secure enough way to open it up for external traffic?

linux security webserver

Share

Improve this question

Follow

edited Oct 24, 2013 at 8:07

icedwater

4,877 ● 3 ● 38 ● 53

asked Aug 7, 2008 at 18:10

karlgrz

14.7k ● 12 ● 48 ● 58

12 Answers

Sorted by: Highest score (default)

5

✓

This article has some of the best ways to lock things down:

http://www.petefreitag.com/item/505.cfm

Some highlights:

- Make sure no one can browse the directories
- Make sure only root has write privileges to everything, and only root has read privileges to certain config files
- Run mod_security

The article also takes some pointers from this book:

Apache Securiy (O'Reilly Press)

As far as distros, I've run Debain and Ubuntu, but it just depends on how much you want to do. I ran Debian with no X and just ssh'd into it whenever i needed anything. That is a simple way to keep overhead down. Or Ubuntu has some nice GUI things that make it easy to control Apache/MySQL/PHP.

Share Improve this answer Follow

answered Aug 7, 2008 at 18:26

helloandre

10.7k ● 9 ● 49 ● 64

5

It's important to follow security best practices wherever possible, but you don't want to make things unduly difficult for yourself or lose sleep worrying about keeping up with the latest exploits. In my experience, there are two key things that can help keep your personal server secure enough to throw up on the internet while retaining your sanity:

1) Security through obscurity

Needless to say, relying on this in the 'real world' is a bad idea and not to be entertained. But that's because in the real world, baddies know what's there and that there's loot to be had.

On a personal server, the majority of 'attacks' you'll suffer will simply be automated sweeps from machines that have already been compromised, looking for default installations of products known to be vulnerable. If your server doesn't offer up anything enticing on the default ports or in the default locations, the automated attacker will move on. Therefore, if you're going to run a ssh server, put it on a non-standard port (>1024) and it's likely it will never be found. If you can get away with this technique for your web server then great, shift that to an obscure port too.

2) Package management

Don't compile and install Apache or sshd from source yourself unless you absolutely have to. If you do, you're taking on the responsibility of keeping up-to-date with the latest security patches. Let the nice package maintainers from Linux distros such as Debian or Ubuntu do the work for you. Install from the distro's precompiled packages, and staying current becomes a matter of issuing the occasional *apt-get update && apt-get -u dist-upgrade* command, or using whatever fancy GUI tool Ubuntu provides.

Share Improve this answer Follow

answered Aug 12, 2008 at 20:33

Dogmang

727 ● 1 ● 7 ● 14

As long as it allows external traffic, it is part of the 'real world' though. – icedwater Oct 24, 2013 at 7:53

▲

2

▼

One thing you should be sure to consider is what ports are open to the world. I personally just open port 22 for SSH and port 123 for ntpd. But if you open port 80 (http) or ftp make sure you learn to know at least what you are serving to the world and who can do what with that. I don't know a lot about ftp, but there are millions of great Apache tutorials just a Google search away.

Share Improve this answer Follow

answered Aug 12, 2008 at 19:52

icco
3,084 ● 4 ● 35 ● 49

2

Bit-Tech.Net ran a couple of articles on how to setup a home server using linux. Here are the links:

[Article 1](#)
[Article 2](#)

Hope those are of some help.

Share Improve this answer Follow

answered Aug 15, 2008 at 22:48

Pondidum
11.6k ● 8 ● 52 ● 69

2

@svrist mentioned EC2. EC2 provides an API for opening and closing ports remotely. This way, you can keep your box running. If you need to give a demo from a coffee shop or a client's office, you can grab your IP and add it to the ACL.

Share Improve this answer Follow

answered Aug 23, 2008 at 22:37

Gary Richardson
16.4k ● 10 ● 54 ● 48

1

Its safe and secure if you keep your voice down about it (i.e., rarely will someone come after your home server if you're just hosting a glorified webroot on a home connection) and your wits up about your configuration (i.e., avoid using root for everything, make sure you keep your software up to date).

On that note, albeit this thread will potentially dwindle down to just flaming, my suggestion for your personal server is to stick to anything Ubuntu ([get Ubuntu Server here](#)); in my experience, the quickest to get answers from whence asking questions on forums (not sure what to say about uptake though).

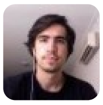
My home server security BTW kinda benefits (I think, or I like to think) from not having a static IP (runs on DynDNS).

Good luck!

/mp

Share Improve this answer Follow

answered Aug 7, 2008 at 18:18



mauriciopastrana

5,070 ● 7 ● 36 ● 36

1

Be careful about opening the SSH port to the wild. If you do, make sure to disable root logins (you can always `su` or `sudo` once you get in) and consider more aggressive authentication methods within reason. I saw a huge dictionary attack in my server logs one weekend going after my SSH server from a DynDNS home IP server.

That being said, it's really awesome to be able to get to your home shell from work or away... and adding on the fact that you can use SFTP over the same port, I couldn't imagine life without it. =)

Share Improve this answer

Follow

answered Aug 7, 2008 at 18:26

saint_groceon

6,227 ● 5 ● 33 ● 26

1

You could consider an EC2 instance from Amazon. That way you can easily test out "stuff" without messing with production. And only pay for the space,time and bandwidth you use.

Share Improve this answer

Follow

answered Aug 7, 2008 at 18:55

svrist

7,110 ● 7 ● 46 ● 67

1

If you do run a Linux server from home, install ossec on it for a nice lightweight IDS that works really well.

[EDIT]

As a side note, make sure that you do not run afoul of your ISP's Acceptable Use Policy *and* that they allow incoming connections on standard ports. The ISP I used to work for had it written in their terms that you could be disconnected for running servers over port 80/25 unless you were on a business-class account. While we didn't actively block those ports (we didn't care unless it was causing a problem) some ISPs don't allow any traffic over port 80 or 25 so you will have to use alternate ports.

Share Improve this answer

Follow

edited Aug 12, 2008 at 20:21

answered Aug 7, 2008 at 21:08

dragonmantank

15.5k ● 22 ● 85 ● 92

1

If you're going to do this, spend a bit of money and at the least buy a dedicated router/firewall with a separate DMZ port. You'll want to firewall off your internal network from your server so that when (not if!) your web server is compromised, your internal network isn't immediately vulnerable as well.

Share Improve this answer

Follow

answered Aug 15, 2008 at 22:33

Carl Russmann

1,719 ● 12 ● 13

0

There are plenty of ways to do this that will work just fine. I would usually jsut use a .htaccess file. Quick to set up and secure *enough* . Probably not the best option but it works for me. I wouldn't put my credit card numbers behind it but other than that I dont really care.

Share Improve this answer

Follow

answered Aug 7, 2008 at 18:12

Adam Lerman

3,399 ● 9 ● 43 ● 53

0

Wow, you're opening up a can of worms as soon as you start opening anything up to external traffic. Keep in mind that what you consider an experimental server, almost like a sacrificial lamb, is also easy pickings for people looking to do bad things with your network and resources.

Your whole approach to an externally-available server should be very conservative and thorough. It starts with simple things like firewall policies, includes the underlying OS (keeping it patched, configuring it for security, etc.) and involves every layer of every stack you'll be using. There isn't a simple answer or recipe, I'm afraid.

If you want to experiment, you'll do much better to keep the server private and use a VPN if you need to work on it remotely.

Share Improve this answer

Follow

answered Aug 7, 2008 at 18:18

Marcel Levy

3,437 ● 1 ● 31 ● 40