

Forum:[Hack Forums](#)

Thread: [\[Tutorial\] Server Firewall hardening](#)

guardian - December 29, 2020, 8:43 am

In order to secure your server, you will have to have a well configured firewall

Instead of blocking stuff you do not want, you should block everything and only allow stuff that you want - also called whitelisting

Warning: Be careful with your configurations, as you might lock yourself out of your server, if not done correctly.

Do not run the commands one after one, instead run them inside a shell script, as otherwise you will be locked out, before you can finish your firewall configuration

First of all, block all traffic using iptables

Code iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

If you would leave your iptables configuration like that, your server would be secure, but you would not be able to access it yourself either - so we have to add exceptions

We assume you have SSH running on port 37282

For your to be able to use ssh, you will have to add an iptables rule that allows traffic on the specific port

Code iptables -A INPUT -i lo -j ACCEPT

iptables -A INPUT -p tcp -m tcp --dport 37282 -j ACCEPT

iptables -A OUTPUT -o lo -j ACCEPT

iptables -A OUTPUT -p tcp --sport 37282 -m state --state ESTABLISHED -j ACCEPT

We also accept I/O on the loopback interface (127.0.0.1), as it is required by the application

So assuming you have SSH running on port 37282, every kind of traffic will be dropped by your box, except the ssh connection on port 37282

To further harden the system you could only allow access to the port for one specified IP

A port knock would be another step, which only opens the port, when a specific set of ports has been connected to by a single IP in a specific order - this will cause a set port to open

If you want a Part 2, where more detailed explanations about measures such as a port knock are provided, let me know

Juoku - December 2, 2021, 11:02 am

This is a great guide! I dunno how this has 0 replies lol

Thanks for sharing mate! this will definitely come in handy, never thought about whitelisting before.

guardian - December 2, 2021, 10:28 pm

(Dec 2, 2021 12:02 PM)Juoku Wrote: [\[CITE\]](#) This is a great guide! I dunno how this has 0 replies lol

Thanks for sharing mate! this will definitely come in handy, never thought about whitelisting before.

Thanks man! And probably because that's not what most people here are interested in haha.

The_Defiant_One - January 5, 2022, 10:25 am

Very informative tutorial buddy. Thanks for sharing!

guardian - January 5, 2022, 10:47 am

(Jan 5, 2022 11:25 AM)The_Defiant_One Wrote: [\[CITE\]](#) Very informative tutorial buddy. Thanks for sharing!

Thanks for the feedback, highly appreciated! Glad you liked it. :)

SIM-4391894 - January 5, 2022, 10:59 am

Thanks for writing this tutorial. But if someone tries to do bruteforce on ssh port?

blackrose_x0x0 - January 5, 2022, 2:09 pm

Another one to add to the bookmarks. Thanks for another awesome guide Guardian!

BooZee - January 5, 2022, 5:04 pm

Great tutorial.

Curious, do you regularly see other ports other than port 22 being used for SSH? I have never seen change the default port in practice.

(Jan 5, 2022 11:59 AM)SIM-4391894 Wrote: [\[CITE\]](#) Thanks for writing this tutorial. But if someone tries to do bruteforce on ssh port? Ideally, assuming you set things up correctly, knowing the target port alone will not matter. Your account (root, ec2-user, etc.) should at least have password protection and can have certificates on top of that. From outside the server itself, you can setup Route Tables (subnet), to only allow certain IP addresses. If you're using AWS you can add security groups and use that.

OP actually mentions part of what I mentioned.

Quote: To further harden the system you could only allow access to the port for one specified IP. A port knock would be another step, which only opens the port, when a specific set of ports has been connected to by a single IP in a specific order - this will cause a set port to open.

@Guardian, this is the first time I heard of the term port knocks. Reading about that now - thanks.

kuthacker - January 5, 2022, 8:59 pm

Great tutorial using it right now!

guardian - January 7, 2022, 4:18 am

(Jan 5, 2022 11:59 AM)SIM-4391894 Wrote: [\[CITE\]](#) Thanks for writing this tutorial. But if someone tries to do bruteforce on ssh port?

You could IP restrict it or use a port knock. That way you can protect it from 3rd parties brute forcing it. Also make sure to use a good password.

(Jan 5, 2022 03:09 PM)blackrosex0x0 Wrote: [\[CITE\]](#) Another one to add to the bookmarks. Thanks for another awesome guide Guardian!

Glad you liked it! Thanks for the kind words. :)

(Jan 5, 2022 03:46 PM)forfeit Wrote: [\[CITE\]](#) a good way to have to literally manually allow every service running on your box lmao

Well what would you recommend to do? It's a lot easier to maintain a ruleset like that than blacklisting everything dangerous. Because you should pretty much consider the whole internet as dangerous.

(Jan 5, 2022 09:59 PM)kuthacker Wrote: [\[CITE\]](#) Great tutorial using it right now! Thanks man! Glad you liked it!

(Jan 5, 2022 06:04 PM)BooZee Wrote: [\[CITE\]](#) Great tutorial.

Curious, do you regularly see other ports other than port 22 being used for SSH? I have never seen change the default port in practice.

Ideally, assuming you set things up correctly, knowing the target port alone will not matter. Your account (root, ec2-user, etc.) should at least have password protection and can have certificates on top of that. From outside the server itself, you can setup Route Tables (subnet), to only allow certain IP addresses. If you're using AWS you can add security groups and use that.

OP actually mentions part of what I mentioned.

@Guardian, this is the first time I heard of the term port knocks. Reading about that now - thanks. Thanks man! And yes, of course. I never use the default port for ssh. For a simple server I change the port + IP restrict it. If I have a server that needs to be hardened more tightly, I will use other measures.

SIM-4391894 - January 7, 2022, 7:33 pm

(Jan 7, 2022 05:18 AM)guardian Wrote: [\[CITE\]](#) You could IP restrict it or use a port knock. That way you can protect it from 3rd parties brute forcing it. Also make sure to use a good password.

Glad you liked it! Thanks for the kind words. :)

Well what would you recommend to do? It's a lot easier to maintain a ruleset like that than blacklisting everything dangerous. Because you should pretty much consider the whole internet as dangerous.

Thanks man! Glad you liked it!

Thanks man! And yes, of course. I never use the default port for ssh. For a simple server I change the port + IP restrict it. If I have a server that needs to be hardened more tightly, I will use other measures.

Sure, but sometimes root password or user passwords of ssh are revealed using laravel dorks in .env file. Then how do you secure from this? what other measures you're going to implement if server needs further hardening?

guardian - January 7, 2022, 10:25 pm

(Jan 7, 2022 08:33 PM)SIM-4391894 Wrote: [\[CITE\]](#) Sure, but sometimes root password or user passwords of ssh are revealed using laravel dorks in .env file. Then how do you secure from this? what other measures you're going to implement if server needs further hardening?

If you IP restrict it or implement port knock, how is the other party going to login? Only your IP will have access or they will have to guess the port knock combination. But apart from that, this shouldn't even happen, if you harden the web application the right way.

I mean of course there is a lot of other measures you will need to take into account when hardening a web server, this tutorial only shows some basic firewall ideas. That's by far not everything that you will need to do.