Subject: Any way to limit SSH bruteforce scanning of VPS's on the node?
Posted by mustardman on Wed, 21 Dec 2011 01:44:32 GMT
View Forum Message <> Reply to Message

Hi,

My VPS's are getting a lot of SSH bruteforce scanning. It's getting to the point where it's adding a significant amount of load to the nodes. Besides using something other than port 22 on each VPS and doing things with iptables on each VPS, is there anything I can do on the node?

I was thinking maybe rate limit port 22 in iptables at the node before the VPS forward statements.

I don't want to do this on a production system though so I was wondering if anyone has successfully done something like this.

The iptables statements I have tried in individual VPS's are:
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --set --name SSH
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --seconds 60 --hitcount 6 --rttl --name SSH -j DROP

It would be easier to do it globally on the node and perhaps there are other benefits to dropping the packets before they get forwarded.

I have a bunch of existing VPS's so I don't want to have to go through all of them and make changes. I'll probably use a different port on new VPS's but I still gotta deal with the existing ones.

Subject: Re: Any way to limit SSH bruteforce scanning of VPS's on the node?
Posted by Ales on Wed, 21 Dec 2011 11:47:28 GMT
View Forum Message <> Reply to Message

I understand that you need a good solution with as little per VPS configuration as possible, but I don't know if that's easily achievable. Custom iptable rules seem the best approach in your case.

That aside, I think the most effective solutions would be changing the sshd port and using fail2ban.

There are other tools similar to fail2ban, like denyhosts and BlockHosts. Last time I made comparisons, fail2ban was the most versatile one and it seems most widely used.

If you go for fail2ban, make sure to patch it's init script on the hardware node, since it interferes with the fail2ban services on the VMs. Patch can be found somewhere in Red Hat's bugzilla, just search for fail2ban.

I'd recommend using both, different sshd port and one of these tools, at the same time. Just changing the port is a trivial obstacle to overcome if someone is targeting your server specifically.

Subject: Re: Any way to limit SSH bruteforce scanning of VPS's on the node?
Posted by mustardman on Wed, 21 Dec 2011 21:49:56 GMT
View Forum Message <> Reply to Message

I'm familiar with fail2ban. It has its uses if you know how to set it up for yourself which I do. I'm not a big fan of setting it up for customers on VPS's though. The process uses too much memory which is at a premium on VPS's. Also it's a bit awkward to set up. Again, not a problem if you are doing it yourself for your own use but not if you want a cookie cutter solution to bang out to customers.

The other solutions same thing. Just adds more complexity. I'm looking for something that keeps it simple and just works. So in my mind that excludes anything that uses lists and processes and log files. Which leaves me with iptables which is already built into the kernel, does not use more memory, does not add much if any overhead, is usually already running.

Not sure what you mean about running fail2ban on the nodes and patching. I don't have problems on the nodes. I change the ssh ports on the nodes and use keys instead of passwords.

Subject: Re: Any way to limit SSH bruteforce scanning of VPS's on the node?
Posted by Ales on Thu, 22 Dec 2011 01:54:18 GMT
View Forum Message <> Reply to Message

I agree, a solid iptables solution that's on the HN side only is the most clean approach. Especially if you don't have complete control over VMs or need to keep resource consumption as low as possible.

I'd start by looking at existing stateful firewalls to see how they limit access to sshd. Just to get some more ideas about possible iptables rules and policies.

I'm afraid I don't have a more specific suggestion than this. Perhaps someone else has done more research in this direction and can chip in.

Just to note, fail2ban's memory consumption is caused by a large default stack size on linux (ie. 10MB on SL6). I believe fail2ban would be quite happy with 256kB but it would need a small patch to put this into effect on SL/CentOS/RHEL. This would lower fail2ban's memory consumption at least tenfold.

It's worth looking into if you are using it on some nodes end would need to lower its memory footprint.

Also, when I mentioned patching its init script (fail2ban from EPEL 6) for openvz, I meant this: currently, when attempting to start the program, its init script simply looks for any running fail2ban processes and if it sees any, it won't start fail2ban.

This is a problem when you try to run fail2ban on the HN while it's already running in some VM,

since this VM process is visible on the HN and the init script incorrectly assumes fail2ban is already running on the HN itself.

There is a bug open about this at EPEL but I don't think they'll act on it since the patch provided is openvz specific. Anyway, a patch is provided in the bug report and it works just fine.

It's a bit unrelated to what you're asking but I thought I'd mention it since you said you use fail2ban too.