Suppose you have a firewall that's supposed to block SSH connections, but instead allows them through. How would you debug it?

I had a similar issue when working on my ELK stack project when I was not able to SSH into my servers, but other local devices were. My intent was to block all SSH connections on all virtual machines that were not my own. The JumpBox is the only machine I can SSH into from my local device and my local device is the only device in which the home IP address is whitelisted for access. I can then SSH into any of my other servers and/or virtual networks from the JumpBox through private or temporary public IP addresses, minimizing access. My JumpBox can access Web 1, Web 2, Web 3, and ELK-SERVER. With my SSH Connection rule in place, all SSH traffic except mine should be denied except those and would then get an error code if they attempted.They are denied access because I have not specified their IP address as allowable in the files, but before I had all of my rules correctly configured, I could not get into any of my servers privately or through the JumpBox, but my servers were accepting all SSH connections due to an error in how I set up my rule. I had the source and destination set as any, so as long as the port and IP addresses were known, anyone could get in, but the port number I had there was incorrect. I fixed my rule to reflect my JumpBox as the source and destination set as any, in addition to correcting the port number.

If the issue was not as straightforward as my ELK project and my firewall was not blocking SSH connections and allowing anyone in when it should not be, I would assume the source of error is very likely somewhere in the security group's rules and I would check that first. The specific pane I would be looking for in Azure would be "Network security groups" and select the group connected with the server that is having issues for further investigation. I would check the security group's inbound security rules and first check that the rule is there, then that the source and destination have the correct filters. If so, I would move on to check if the port is correct and then check that it can only be accessed by those specified. I would also check my files to ensure that "remote_user" has the correct username and that all IP addresses are correct in my files as well. If anything needs to be changed, it will be done promptly. To test that my changes are working correctly, I will then attempt to SSH from a blocked address and will know I have been diligent when I receive an error. I would also use the command **systemctl** to check the status of my SSH service and the command **sudo ufw** to block unwanted SSH connections, but since I need my specific SSH connections, this usually stays as allowable and other security measures are taken place in the network security group in Azure. I can never assume my network is 100% and always immune to all unauthorized attempts and access, but it

gives me some peace of mind and I can add many layers of security and protection to make it as difficult and annoying as possible to whomever might try to penetrate my network. Kibana is an amazing tool that I actively use to monitor my servers and networks to prevent any malicious attempts or other suspicious activity. I would also set up several alerts to notify my system if anything were to be attempted, such as tons of SSH login attempts in a short span of time or high traffic from an unwarranted location.