**Network Security**

The problem stated is that a server is running HTTP on port 80 even though compliance guidelines require encryption. In project 1 we did have servers running on Port 80. This is because we were establishing a connection from our own and known home IP address, to the virtual network we created. There is not much of a security risk when using it for virtual environments that you yourself have created and do not intend on sharing. In an actual deployment I would configure the jump box provisioner to need to use HTTPS protocol instead. Running HTTP on port 80 is a potential problem because it leaves the user vulnerable to man in the middle attacks. To safely use HTTP protocol it would be best practice to keep port 80 open and use 301 redirects to change the traffic to HTTPS. This enables traffic to move securely. To do this one would not necessarily need more tools, they would just need to write a file that makes the connection go through a 301 redirect. Once written and implemented there should not be any extra work to keep the redirect ongoing, and no users should see any breaks.

**Cloud Security**

To control access to a cloud network one needs to restrict user access by making them sign in with a username and password or another type of key. In project 1 we used a cloud network that was only accessible by SSHing into our JumpBox provisioner, there were not a lot of access controls to configure as we were the only ones using the virtual environment. That being said there is the option of viewing access control and adding another user who has different access capabilities than us (the owner), this is relevant because in a real world scenario there is a good chance that one will be working with an environment that has multiple different users who all have different levels of access. One of the access controls we did use was a Network Security Group, we set up inbound and outbound rules for our red team virtual network along with implementing allow/deny actions within the security group. For example one of network security rules is to allow inbound and outbound traffic to the virtual network using any port or protocol. Another example was our connection from our jump box provisioner to our virtual network. This access was restricted to SSH key login, these types of controls were necessary for the project in order to teach us how to harden a virtual environment. My network security group had a couple different rules, not just the SSH from the jump box. Another one was the rule to SSH from our home IP and allow the connection to the jump box we created. The jump box only works if the SSH key we generated and saved as the correct token is entered. We also set up a load balancer and allowed it inbound. The jump box server is scalable up to a point. The Jump box becomes a choke point when an increasing number of people need additional performance from the system, as it is basically just a lone server. For what we were doing I do not think there is a “better” solution but there are alternatives such as a VPN. One disadvantage of implementing a VPN though is the possibility of connection breaks while using it; which then would result in the user being kicked out of the system and back to their normal connection. An advantage of a VPN would be the ability to access the servers without having to go through an SSH prompt and login. The VPN would allow an individual to login through it which then grants them access to whatever they are attempting to connect to, like company servers etc… It is appropriate to use a VPN when a large company wants to keep its data secure but also allows its employees to access the information remotely and securely.

**Logging and Monitoring**

There are a couple ways to determine if a security event or alert is important enough for escalation. For example the data we looked at in our Kibana activity; the event I viewed was almost double all the events that occurred in the same week. This could potentially indicate that someone is trying to gain access to a particular system or exploit it. This activity is what led me to investigate it further. Upon diving deeper I noticed that there was an RPM file downloaded. This is a Red Hat Package manager that is used to store installation packages on systems running Linux. This is something one would want to escalate because having an RPM show up from a place that has double the events than everything else for the week could be a sign of unsavory intent from the user. Looking at the location of the questionable data is also helpful, using Kibana in this sense is perfect. For example the data I used had its source location coming from India but the things being accessed were in China. I used Kibana to look at a number of things relating to this activity. Kibana was able to provide me with the source IP of 35.143.166.159; as well as the geo coordinates { "lat": 43.34121, "lon": -73.6103075 }. It also told me the OS the machine was running as well. I do not think the metric beat file that was downloaded intended to be malicious, at least in the beginning. It seems the user was using it to gather info which then could be used to exploit systems or people. I think more data from the user over a longer period of time would be useful in determining my conclusion. Especially if the user was laying dormant for a while then started their process. It may also be useful to track the users activity for sometime after the event to see if it happens again.