# Identify a file or resource that can be used by Splunk for getting login information. Give both the name of the file and path.

source=/var/log/auth.log

# What other files can be found in the path and for what?

In /var/log, other log files can be found such as boot.log to show startup info and their success status, dpkg.log to show package history such as when a package was installed, unpacked or removed, and dmesg to show hardware status

# Setup ingestion for the file in question 1 into Splunk

To ingest the auth.log file you would go into Add Data and upload the file into Splunk

# Make a report of login/logout attempts and the session duration. Use the command rex to extract fields.

User\_Name=\* | rex "action=(?<newaction>\w+)" | search newaction=login OR action=logout | table

\_time, newaction, User\_Name, stats(Session\_Duration), status

# Find if a session went into admin rights. You may use eval to create a new column using if.

| eval Diff=if(Action="login", 1, if(Action="logout", -1, 0)) | stats sum(Diff) as openSessions by User\_Name | timechart max(openSessions) by User\_Name

# Find the average duration of a session, and the average user session duration. For this you will need to use stats avg.

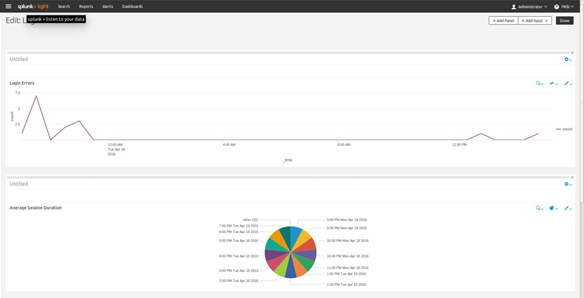
Session\_ID =”\*” | stats range(time) AS Session\_Duration\_sec by SessionID | stats avg(Session\_Duraion)

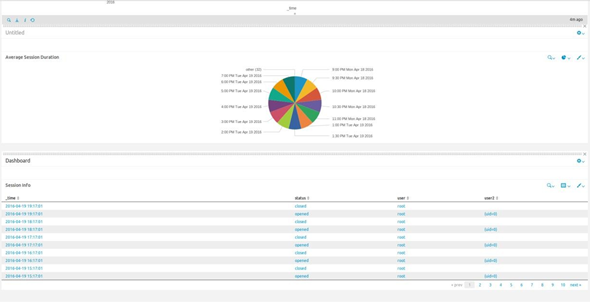
Session\_ID=”\*” | stats avg(duration) min(duration) max(duration) stdev(duration) by User\_Name

# Make a dashboard which has a graph of: average session duration time chart, login errors time chart, event table with session info (\_time, user name, successful?, went into root?). Provide a screenshot of the dashboard and the queries used to generate each of the parts.

host="rob-VirtualBox" source="/var/log/auth.log" authentication failure | timechart count host="rob-VirtualBox" source="/var/log/auth.log" | rex field=\_raw "session closed for user (?<user>[^\s]+)" | search user=rob user=rob | join type=inner usetime=true earlier=true max=1 user[search host="rob-VirtualBox" source="/var/log/auth.log" | rex field=\_raw "session opened for user (?<user>[^\s]+)"] | timechart avg(\_time) by user

host="rob-VirtualBox" source="/var/log/auth.log" ": session" | rex field=\_raw ": session (?<status>[^\s]+) for user (?<user>[^\s]+)" | join type=inner user[host="rob-VirtualBox" source="/var/log/auth.log" ": session" user by | rex field=\_raw ": session (?<status>[^\s]+) for user (?<user>[^\s]+) by (?<user2>[^\s]+)"]





# Explain what settings or information you could use from this dashboard to create alerts to detect potential intruders. Which IPtables command(s) would you use to react if the alert fires? Assume that you can get all the information needed from the log file event.

The information from this dashboard tells me that I could create an alert based on excessive failed login attempts to root or another user. We could also check for session duration times that are longer than expected, or create alerts based on login attempts during unexpected hours of the day. We could further tie this in with if the connection was from a new IP. We could block specific ports from the IP if we don't want to completely block the connection too.

#block all communication from offending IP

sudo iptables -A INPUT -s 12.34.56.78 -j DROP

#drop attempts to port but still allow access

sudo iptables -A INPUT -s 12.34.56.78 -p tcp --destination-port 25 -j DROP

# Use IPtables MAC address filter to allow login only from specific local machines. You may write any valid/non-valid MAC address in your command.

#Default policy set to drop all incoming packets sudo iptables -P INPUT DROP

#Allow input from specific mac address

sudo iptables -A INPUT -m mac --mac-source 00:01:02:03:04:05 -j ACCEPT

# Specify a logrotate that could handle this file to rotate every day, keeping only one week of files. Anything older must be zipped into a tar file. Once past one month it should be deleted

{

daily rotate 1

compressext .tar maxage 30

}