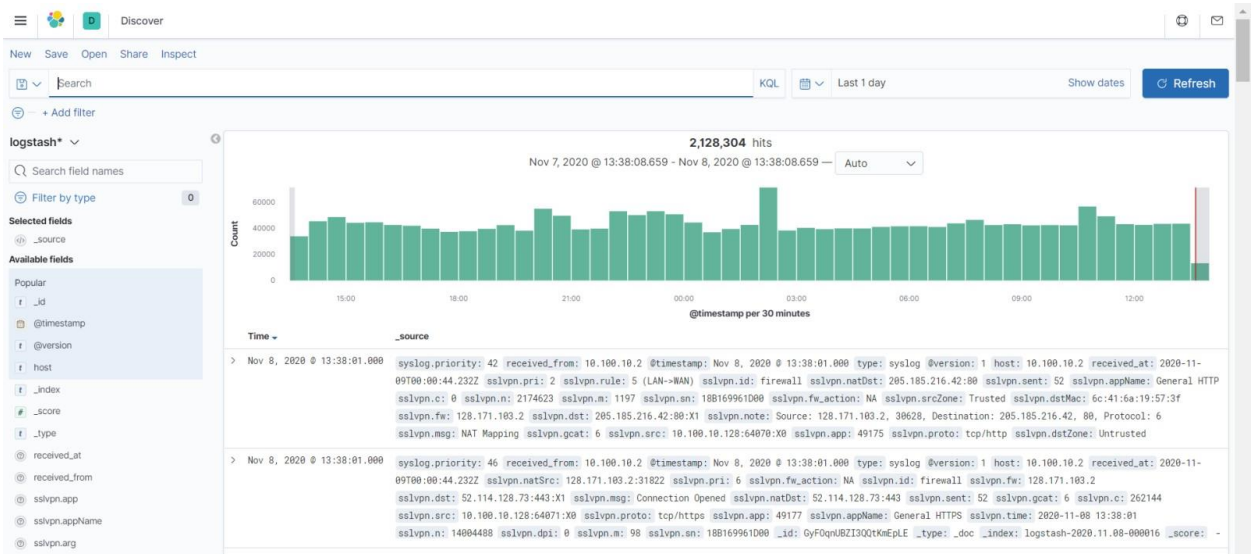
ELK-Stack Reporting Guide

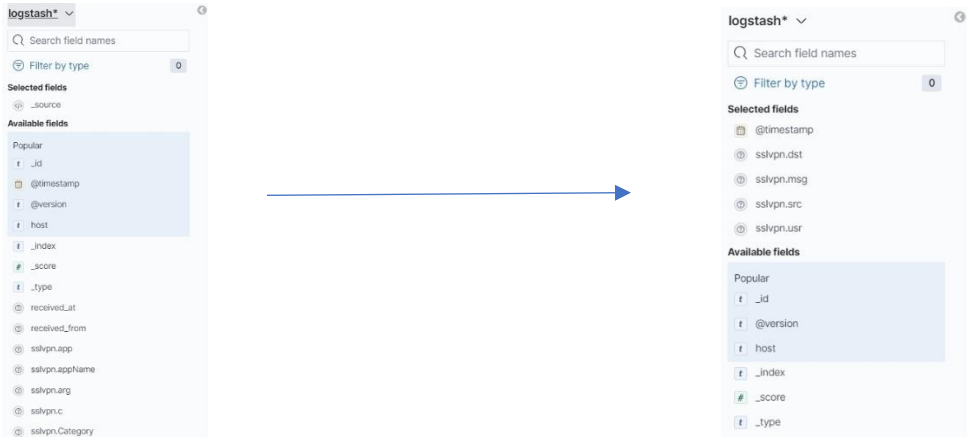
Forward: In this guide I will outline how to create a scheduled report for the ELK-Stack. I will go through steps with the assumption that Elasticsearch, Logstash, and Kibana are already installed on your system. This guide also assumes the ELK-Stack is installed on a CentOS 7 operating system.

Step 1 – Create and save a reusable search based on fields:

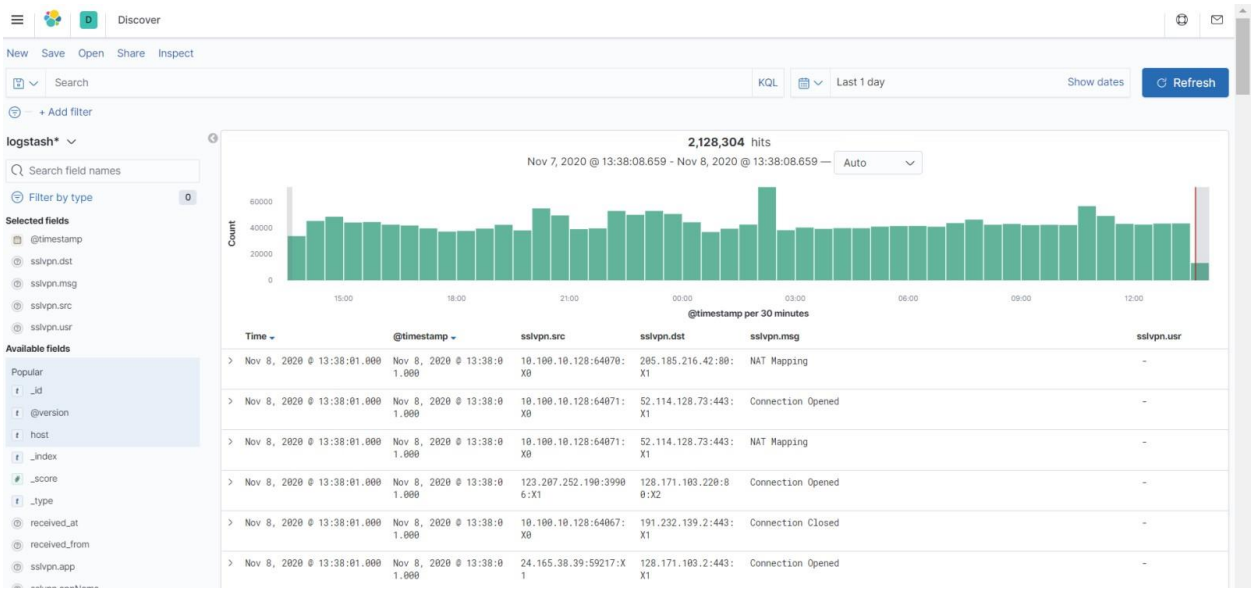
First navigate to Discover in Kibana and toggle to the Index Pattern you will be working with (logstash* in this example).



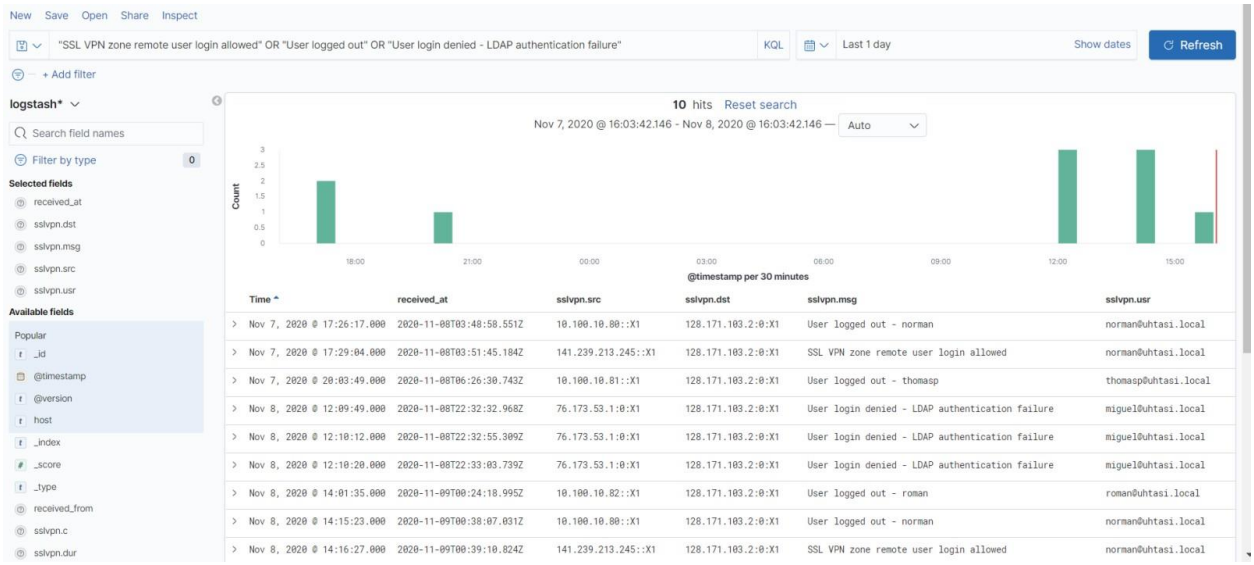
Next Add from Available fields the fields you want to include in your report. These are the data fields you want to keep and allows you to exclude data that is of no use to your report.



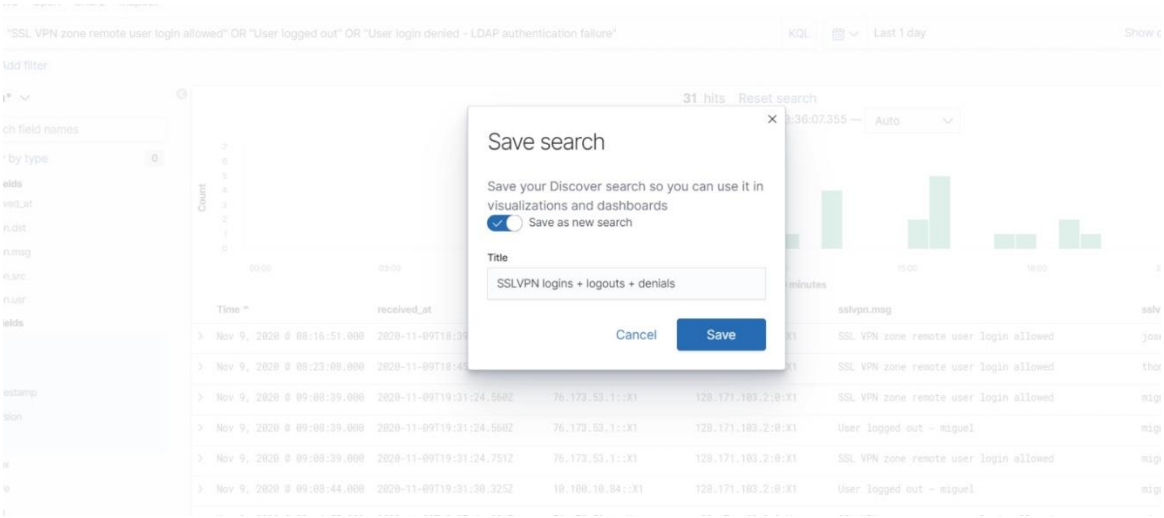
As you can see we added `@timestamp`, `sslvpn.dst`, `sslvpn.msg`, `sslvpn.src`, and `sslvpn.usr` from Available fields to Selected fields. This is the new Discover display with only the Selected fields:



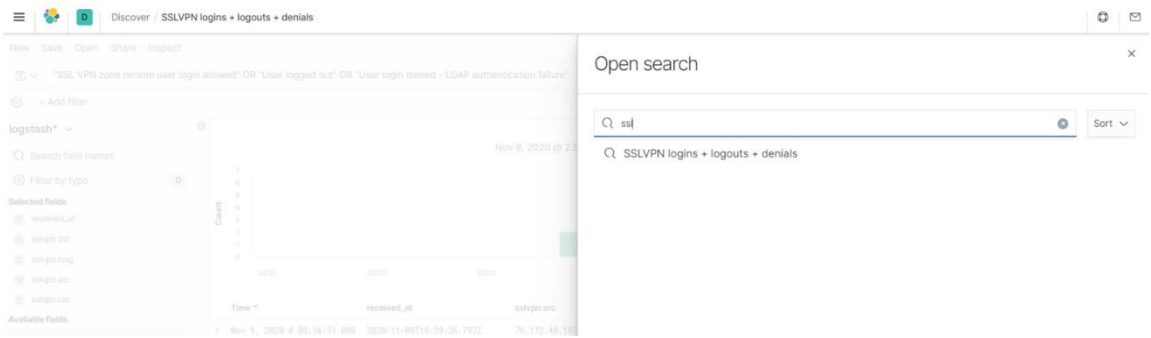
Next in the Search bar, we can specify key search terms that will only display logs with the terms in the Search bar. You can use the terms in combination with connectors like AND or OR to show combinations of logs that contain the term(s). Here is an example that uses the search string `"SSL VPN zone remote user login allowed" OR "User logged out" OR "User login denied - LDAP authentication failure"` to only display SSLVPN login attempts:



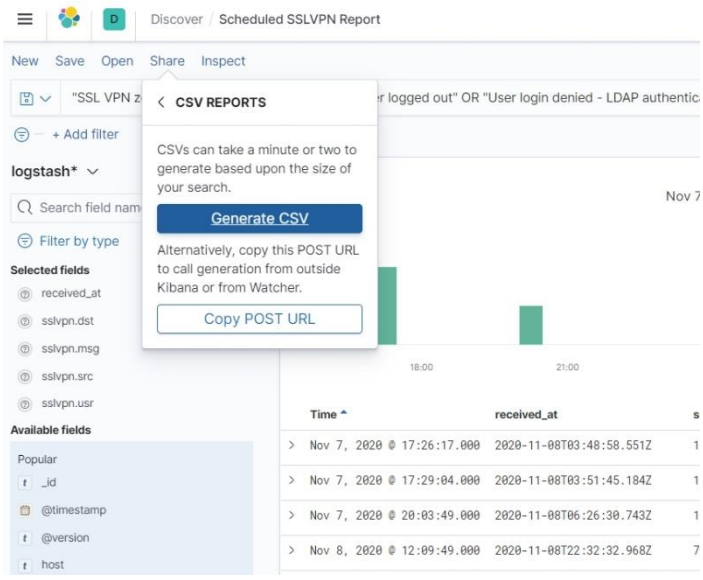
Lastly, we can save the settings we chose (the Selected Fields and Search terms) so that we can reload them at any time. Do this by clicking Save above the Search bar.



Now in the future you can load saved searches by clicking Open next to Save and then searching for the search in the pop-up:



Step 2 – Obtain and edit the POST URL for use in reporting:



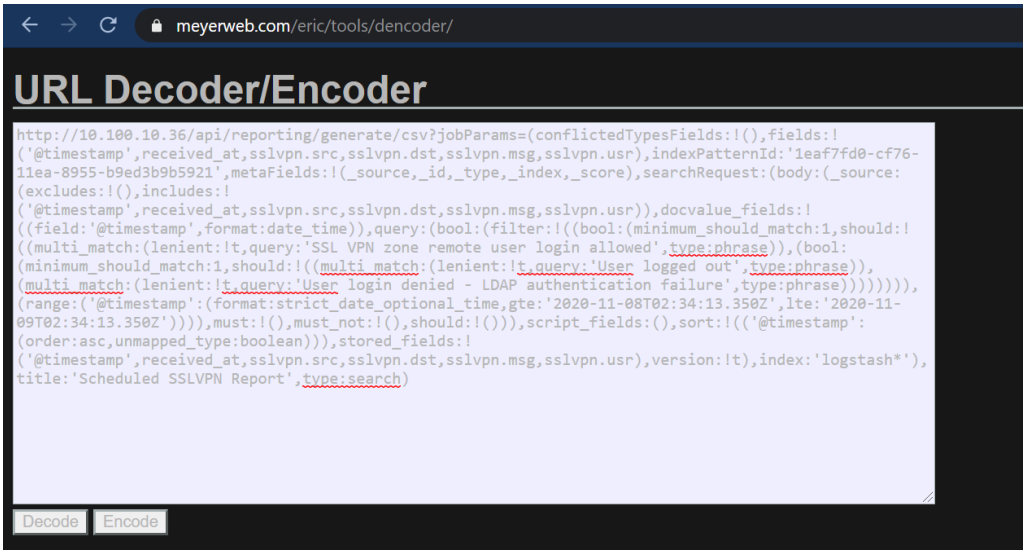
Click Share > CSV Reports > Copy POST URL to copy the generated POST URL. It is necessary to obtain the POST URL for report generation outside of Kibana. In our case, the POST URL will be used in a bash shell script to obtain the report each time the shell script is executed.

Here is an example of a POST URL:

```
http://10.100.10.36/api/reporting/generate/csv?jobParams=%28conflicted
TypesFields%3A%21%28%29%2Cfields%3A%21%28%27%40timestamp%27%2Creceived
_at%2Csslvpn.src%2Csslvpn.dst%2Csslvpn.msg%2Csslvpn.usr%29%2CindexPatt
ernId%3A%271eaf7fd0-cf76-11ea-8955-
b9ed3b9b5921%27%2CmetaFields%3A%21%28_source%2C_id%2C_type%2C_index%2C
_score%29%2CsearchRequest%3A%28body%3A%28_source%3A%28excludes%3A%21%2
8%29%2Cincludes%3A%21%28%27%40timestamp%27%2Creceived_at%2Csslvpn.src%
2Csslvpn.dst%2Csslvpn.msg%2Csslvpn.usr%29%29%2Cdocvalue_fields%3A%21%2
8%28field%3A%27%40timestamp%27%2Cformat%3Adate_time%29%29%2Cquery%3A%2
8bool%3A%28filter%3A%21%28%28bool%3A%28minimum_should_match%3A1%2Cshou
ld%3A%21%28%28multi_match%3A%28lenient%3A%21t%2Cquery%3A%27SSL%20VPN%2
0zone%20remote%20user%20login%20allowed%27%2Ctype%3Aphrase%29%29%2C%28
bool%3A%28minimum_should_match%3A1%2Cshould%3A%21%28%28multi_match%3A%
28lenient%3A%21t%2Cquery%3A%27User%20logged%20out%27%2Ctype%3Aphrase%2
9%29%2C%28multi_match%3A%28lenient%3A%21t%2Cquery%3A%27User%20login%20
denied%20-
%20LDAP%20authentication%20failure%27%2Ctype%3Aphrase%29%29%29%29%29%2
9%29%29%2C%28range%3A%28%27%40timestamp%27%3A%28format%3Astrict_date_o
ptional_time%2Cgte%3A%272020-11-
08T02%3A34%3A13.350Z%27%2Clte%3A%272020-11-
09T02%3A34%3A13.350Z%27%29%29%29%29%2Cmust%3A%21%28%29%2Cmust_not%3A%2
1%28%29%2Cshould%3A%21%28%29%29%29%2Cscript_fields%3A%28%29%2Csort%3A%
21%28%28%27%40timestamp%27%3A%28order%3Aasc%2Cunmapped_type%3Aboolean%
29%29%29%2Cstored_fields%3A%21%28%27%40timestamp%27%2Creceived_at%2Css
lvpn.src%2Csslvpn.dst%2Csslvpn.msg%2Csslvpn.usr%29%2Cversion%3A%21t%29
%2Cindex%3A%27logstash%2A%27%29%2Ctitle%3A%27Scheduled%20SSLVPN%20Repo
rt%27%2Ctype%3Asearch%29
```

As you can see it is heavily encoded. Before inserting the POST URL into our bash shell script, we must adjust the range field in the POST URL so that the time range is fluid not fixed. This ensures that the POST URL grabs logs based on new dates rather than just a single date range stuck in the past. To do this we can use an online decoder/encoder tool:

https://meyerweb.com/eric/tools/dencoder/



This is the range decoded:

```
(range:('@timestamp':(format:strict_date_optional_time,gte:'2020-11-08T02:34:13.350Z',lte:'2020-11-09T02:34:13.350Z'))))
```

We want the gte and lte time to be fluid and not fixed on a static range. To do this we add “date math” in place of the real times. This is what the new range looks like with “date math”:

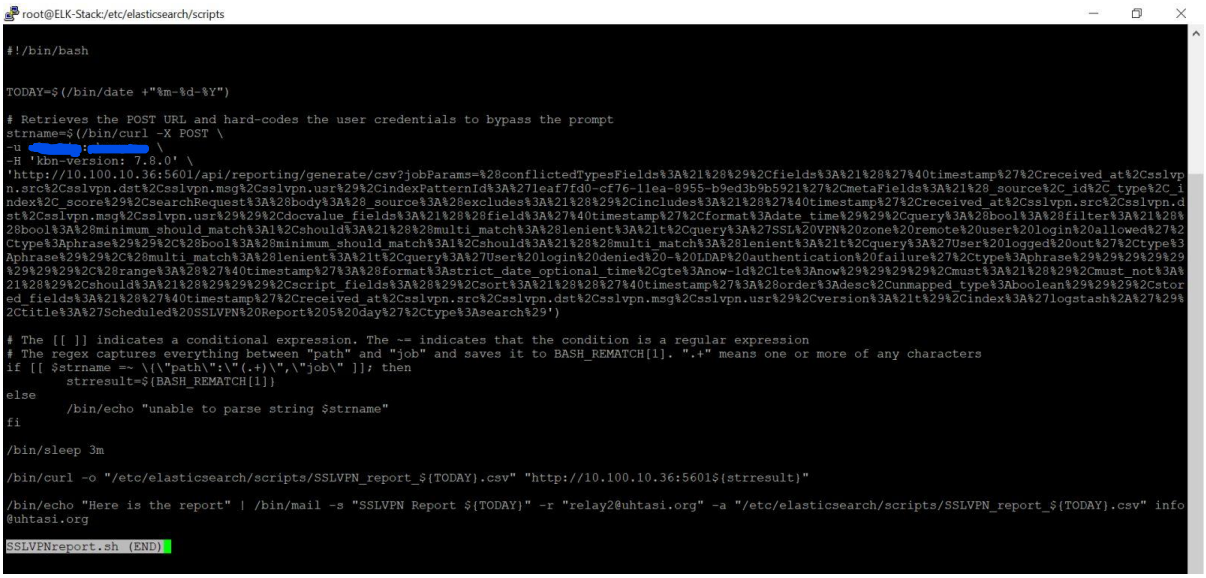
```
(range:('@timestamp':(format:strict_date_optional_time,gte:now-1d,lte:now))))
```

This will cause the POST URL to pull data from the moment it is requested to a time 24 hours before it is requested, which is desired. After changing the range field, do not forget to re-encode the POST URL.

***Notice: We added SSL encryption to our Kibana which required a change in the POST URL. Upon generating the new POST URL it is required to change the path to http://10.100.100.36:5601 not the https://10.100.100.36 which is generated. If this is left unchanged the string will not be able to be parsed.**

Step 3 – Create the bash shell script:

We will create a shell script in the CentOS 7 virtual machine to pull the report in .csv form via the POST URL. We will then email the report to designated addresses after the report completes its download.



This is the script we are using for the SSLVPN report. The only change that needs to be made if using a new search would to be to insert the new POST URL where the current one is. Also you may need to change the kbn-version if the version differs from 7.8.0. Lastly, if you changed the

default elastic:changeme credentials, that would have to be updated as well. The other aspects of the script are commands that allow the report to be emailed to the designated recipient.

Step 4 – Add the script to crontab to automate its execution:

Adding the script to crontab allows us to automate the script and run it on a schedule. This will let us run the script without having to do it manually, and let us obtain the report via email on a scheduled basis. Here is the crontab syntax for our SSLVPN example:

```
00 00 * * * /bin/bash /etc/elasticsearch/scripts/SSLVPNreport.sh
```

The formatting here causes the SSLVPNreport.sh script to execute once a day at 12:00 AM.