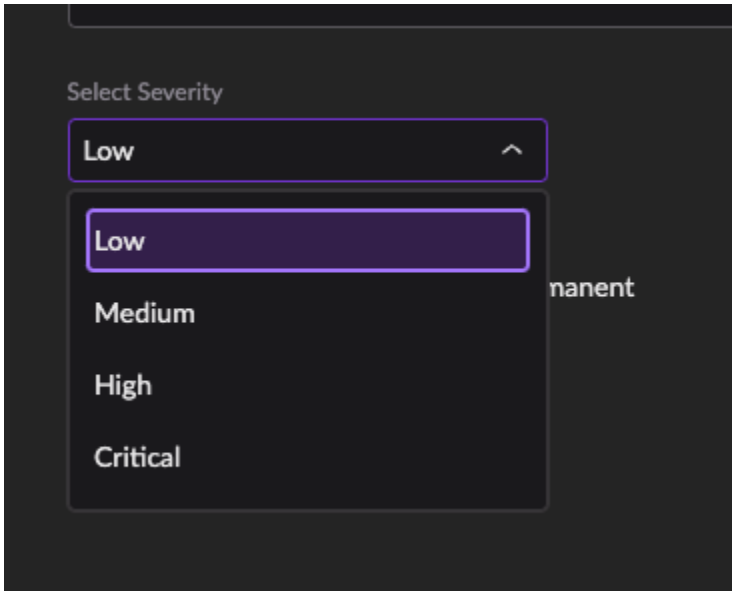
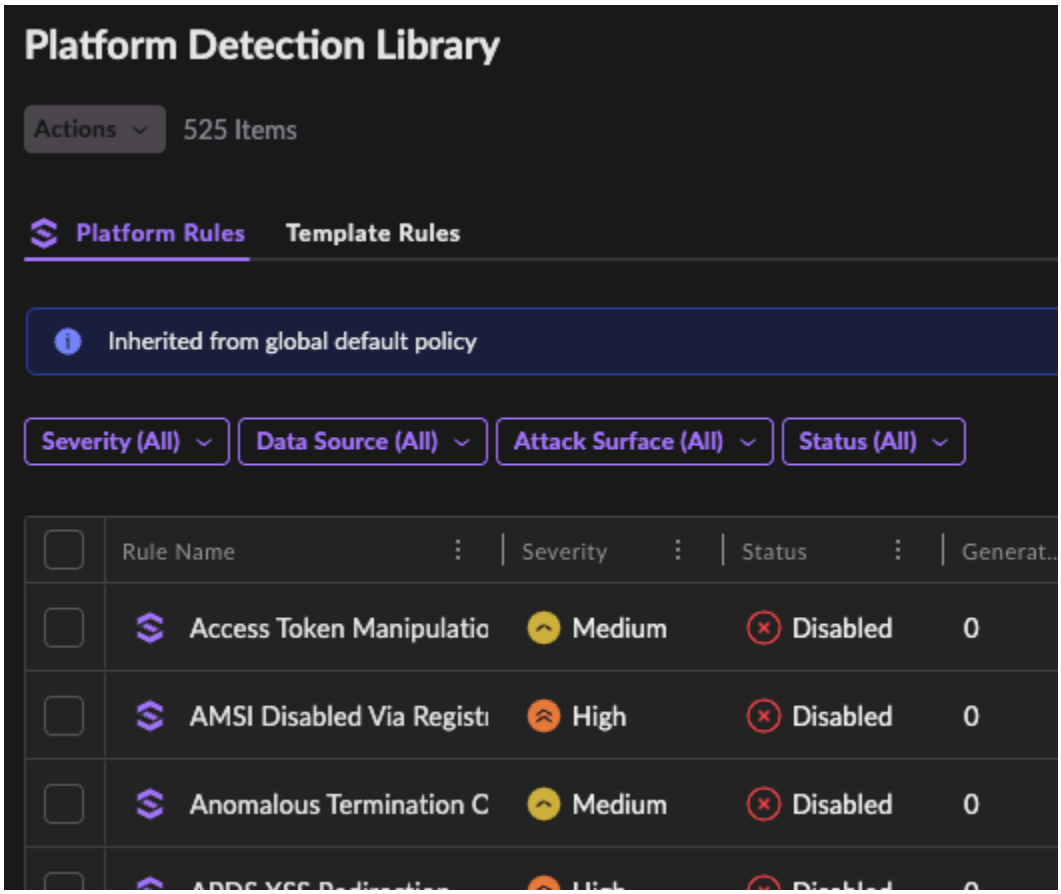
**What Risk Scoring exists within the existing S1 console:**

When creating a STAR rule, you are able to set a severity level




And in the new platform detection library, rules provided by S1 have the same severity score



These values show up in the raw alert logs / data lake as ‘severity_id’ in the dataSource.name=’alert’ data set. The value is set in a range from 1 to 5 (5 being critical). 1 is “info” which is not an option in the rule creation drop down but is used for certain alerts from 3rd parties or S1 correlations.

This is a very simple version of risk scoring by various possible characteristics which are likely to be common across alert types:

- Endpoint / hostname
- IP
- Email
- Username

In order to do this, we must first create a score for each alert. The most basic example of this is to just assign a 1 point per alert. In the S1 Data Lake, there may be many types of alerts to work with

- EDR
- S1 Identity
- S1 Cloud
- Third Party Alerts Logs (email provider, NGFW, etc)

Across these alert types/source there are many fields in common. However there are some that are unique so we will first combine these using ternary operators to unify them 1 field each for endpoint, ip, email, etc

```
Unset
//this field is always populated for S1 alerts so we can just search for its
existence
finding_info.desc=*

//combining multiple possible field names for the hostname/endpoint name into
one field
| let combined.hostname.field = (len(evidences\[0\].src_endpoint.hostname)>4) ?
evidences\[0\].src_endpoint.hostname : resources\[0\].name

//combining multiple fields for ip in to one
| let combined.ip.field = evidences\[0\].src_endpoint.ip

//combining multiple fields for email into one
```

```
| let combined.email.field = evidences\[0\].actor.user.email_addr

//creating a table with one row per endpoint with 1+ alert finding. the table
will create a score (weighted sum of all the severity_id scores for each alert
for the endpoint), a list of all the unique alerts, and all the unique data
sources
| group 'Total Endpoint Score'=sum(severity_id), 'Unique Alerts by Endpoint' =
array_agg_distinct(finding_info.desc), 'Unique Data
Sources'=array_agg_distinct(metadata.product.name) by
Endpoint=combined.hostname.field

//sort in descending order because we want to see the biggest offenders
| sort -'Total Endpoint Score'
```

This particular query is based on the endpoint. But as we have created combined fields for ip, email, etc then you can flip this around to do the scoring based on those attributes by just changing the grouping statement.

For example swapping in combined.ip.field instead of endpoint

```
Unset
//this field is always populated for S1 alerts so we can just search for its
existence
finding_info.desc=*

//combining multiple possible field names for the hostname/endpoint name into
one field
| let combined.hostname.field = (len(evidences\[0\].src_endpoint.hostname)>4) ?
evidences\[0\].src_endpoint.hostname : resources\[0\].name

//combining multiple fields for ip in to one
| let combined.ip.field = evidences\[0\].src_endpoint.ip

//combining multiple fields for email into one
| let combined.email.field = evidences\[0\].actor.user.email_addr

//filter out rows which do not have an IP field
| filter combined.ip.field=*
```

```
//creating a table with one row per IP with 1+ alert finding. the table will
create a score (weighted sum of all the severity_id scores for each alert for
the endpoint), a list of all the unique alerts, and all the unique data sources
| group Total.IP.Score=sum(severity_id), 'Unique Alerts by IP' =
array_agg_distinct(finding_info.desc), 'Unique Data
Sources'=array_agg_distinct(metadata.product.name) by IP=combined.ip.field

//sort in descending order because we want to see the biggest offenders
| sort -Total.IP.Score
```

In this case you can see that I filtered to only those results which have the IP field as a fair number of the alerts just didn’t have one.