Increase in source code downloads

You might need to find users who have downloaded more files from Git than normal when doing the following:

- Monitoring use of Git repositories

Prerequisites

In order to execute this procedure in your environment, the following data, services, or apps are required:

- Splunk Enterprise or Splunk Cloud Platform
- Code management data

Example

Your organization uses Git to version source code. Developers only have access to repositories they need for their projects, but even good access restrictions can't prevent all data exfiltration. You want to monitor Git access for downloads so you can validate that downloads are normal or identify download activity that needs to be investigated.

To optimize the search shown below, you should specify an index and a time range. In addition, this sample search uses atlassian-bitbucket as a source. You can replace this source with any other web server data used in your organization.

1. Run the following search:

```bash
source=/*/atlassian-bitbucket-access.log
|bucket _time span=1d
|stats count BY user _time
|stats count AS num_data_samples max(eval(if(_time >= relative_time(maxtime, "-1d@d"), 'count',null))) AS count
|stats count AS num_data_samples max(eval(if(_time<relative_time(maxtime, "-1d@d"), 'count',null))) AS count
|stats count AS num_data_samples max(eval(if(_time<relative_time(maxtime, "-1d@d"), 'count',null))) AS count
|stats count AS num_data_samples max(eval(if(_time<relative_time(maxtime, "-1d@d"), 'count',null))) AS count
|stats count AS num_data_samples max(eval(if(_time<relative_time(maxtime, "-1d@d"), 'count',null))) AS count
|eval lowerBound=(avg-stdev*2), upperBound=(avg+stdev*2)
|where 'count' > upperBound AND num_data_samples >=7
```

Search explanation

The table provides an explanation of what each part of this search achieves. You can adjust this query based on the specifics of your environment.
### Splunk Search

```
source="*/atlassian-bitbucket-access.log"
```

<table>
<thead>
<tr>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pull in your Git dataset.</td>
</tr>
</tbody>
</table>

```
|bucket _time span=1d
```

<table>
<thead>
<tr>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect all events occurring in a one-day span.</td>
</tr>
</tbody>
</table>

```
|stats count BY user _time
```

<table>
<thead>
<tr>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count the downloads and aggregate them per user, per day.</td>
</tr>
</tbody>
</table>

```
|stats count AS num_data_samples max(eval(if(_time >= relative_time(maxtime, "-1d@d"), 'count',null))) AS count
avg(eval(if(_time<relative_time(maxtime,"-1d@d"),'count ','null'))) AS avg
stdev(eval(if(_time<relative_time(maxtime,"-1d@d"),'count','null'))) AS stdev BY user
```

<table>
<thead>
<tr>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate the mean, standard deviation, and most recent value.</td>
</tr>
</tbody>
</table>

```
|eval lowerBound=(avg-stdev*2),
upperBound=(avg+stdev*2)
```

<table>
<thead>
<tr>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate the bounds as a multiple of the standard deviation.</td>
</tr>
</tbody>
</table>

```
|where 'count' > upperBound AND num_data_samples >=7
```

<table>
<thead>
<tr>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return results where the count is greater than the upper bound.</td>
</tr>
</tbody>
</table>

## Result

While there are no traditional false positives in this search, there will be a lot of noise, based on the bursty nature of source code access. When someone first clones a repository, their counts will be excessive. This search provides contextual data to record when these big bursts of activity occur.

When this search returns values, initiate your incident response process and validate the user account accessing the specific repos. Contact the user and their manager to determine if it is authorized, and make a note if it is authorized and by whom. If not, the user credentials may have been used by another party and additional investigation is warranted as repositories hold sensitive source code.

Run this search once a day, preferably overnight. If you want to run this search more frequently, or if this search is too
slow for your environment, use a summary index that first aggregates the data.

It is particularly interesting to correlate this behavior to a watchlist that contains the user IDs of personnel that are considered higher risk: contractors, new employees, employees that never go on vacation, and employees with access to particularly sensitive source code.