First time accessing an internal Git repository

You might need to find users who accessed a Git repository for the first time 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

Developers are often granted access to the Git (or other software life cycle repository) that their responsibilities require. The first time a user accesses a given repository could be perfectly normal, or if the repository contains code not relevant to the developer's role, could be an anomaly to investigate. You want to monitor first access instances so you can investigate if needed.

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:

```
source="*/atlassian-bitbucket-access.log"
|rex "GET /projects/[\^]*/repos/(?<git_repo>[\^]*)"
|rex "(?<git_repo>[\^]*)\.git"
|rex "git:[\^ ](1,)/projects/[\^]*/repos/(?<git_repo>[\^]*)"
|search git_repo="*"
|stats earliest(_time) AS earliest latest(_time) AS latest BY user, git_repo
|where earliest > relative_time(now(), "-1d@d")
```

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.
<table>
<thead>
<tr>
<th>Splunk Search</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>source=&quot;*/atlassian-bitbucket-access.log&quot;</td>
<td>Pull in your Git dataset. Your may be different from the Atlassian sample shown here. Update the source according to your environment.</td>
</tr>
<tr>
<td>/ rex &quot;GET /projects/[^/]<em>/repos/(?&lt;git_repo&gt;[^/]</em>)&quot;</td>
<td>Extract the field names. These regular expressions have worked in a couple of environments, but you may need to adapt them to yours.</td>
</tr>
<tr>
<td>/ rex &quot;(?&lt;git_repo&gt;[^/]*).git&quot;</td>
<td></td>
</tr>
<tr>
<td>/ rex &quot;git.[^ /]{1,}/projects/[^/]<em>/repos/(?&lt;git_repo&gt;[^/]</em>)&quot;</td>
<td></td>
</tr>
<tr>
<td>/ search git_repo=&quot;*&quot;</td>
<td>Filter for logs that include a git_repo field.</td>
</tr>
<tr>
<td>/stats earliest(_time) AS earliest latest(_time) AS latest BY user, git_repo</td>
<td>Calculate the earliest and the latest time that the repos were accessed.</td>
</tr>
<tr>
<td>/where earliest &gt; relative_time(now(), &quot;-1d@d&quot;)</td>
<td>Return results where the earliest time was within the last day.</td>
</tr>
</tbody>
</table>

**Result**

While there are no traditional false positives in this search, every time this fires, it will accurately reflect the first occurrence in the time period you're searching over (or for the lookup cache feature, the first occurrence over whatever time period you built the lookup). You should not review these alerts directly (except for access to extremely sensitive repositories), but instead use them for context, or to aggregate risk.

When this search returns values, initiate your incident response process and identify the user account accessing the specific repo. Contact the user and manager to determine if they are accessing the repo with authorization. If they did not access this repo, attempt to determine if the user credentials have been used by another party by stealing a user's credentials.