# Nightwatch Cybersecurity

Cybersecurity services and research

# GitBleed – Finding Secrets in Mirrored Git Repositories – CVE-2022-24975

February 11, 2022April 25, 2022 nightwatchcyber Research, Tools git, gitbleed, github, gitlab

## Summary

Due to a discrepancy in Git behavior, partial parts of a source code repository are visible when making copies via the "git clone" (https://git-scm.com/docs/git-clone) command. There are additional parts of the repository that only become visible when using the "–mirror" option. This can lead to secrets being exposed via git repositories when not removed properly, and a false sense of security when repositories are scanned for secrets against a cloned, non-mirrored copy.

Attackers and bug bounty hunters can use this discrepancy in Git behavior to find hidden secrets and other sensitive data in public repositories.

Organizations can mitigate this by analyzing a fuller copy of their repositories using the "–mirror" option and remove sensitive data using tools like BFG (https://rtyley.github.io/bfg-repo-cleaner/) or git-filter-repo (https://github.com/newren/git-filter-repo) (which do a more thorough job).

MITRE assigned CVE-2022-24975 (https://nvd.nist.gov/vuln/detail/CVE-2022-24975) to track this issue.

***[ADDED APRIL 2022: This bug is NOT the same as NotGitBleed (https://www.notgitbleed.com/) – see their website here]***

## Technical Details

Git is a popular open source tool used for version control of source code. When users make a copy of a local or remote git repository, they use the "git clone" (https://git-scm.com/docs/git-clone) command. However, this command doesn't copy all of the data in the originating repository such as deleted branches and commits. On the other hand, there is a "–mirror" option which copies more parts of the repository. The discrepancy between the two behaviors can lead to secrets and other sensitive data lingering in the original repository. Additionally, existing tools for secrets detection often operate on cloned repositories and do not detect secrets in the mirror portion of the repository unless cloned via the "–mirror" command.

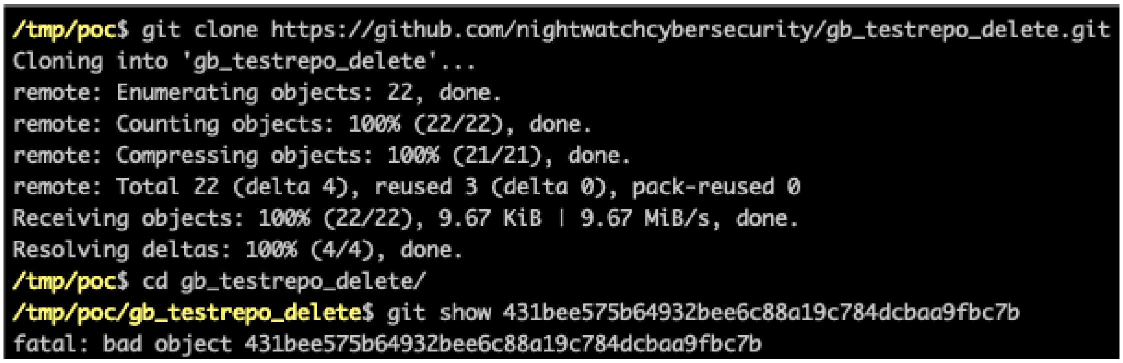
We also tested forking in GitHub and GitLab, and in both systems forking uses the regular "git clone" behind the scenes and not the "–mirror" version. That means that repositories containing secrets in the mirrored portion will not propagate those secrets to their forks.

MITRE assigned CVE-2022-24975 (https://nvd.nist.gov/vuln/detail/CVE-2022-24975) to track this issue.

We provide two examples of repositories containing hidden secrets that are only visible when cloning with the "–mirror" option. These can be found here:

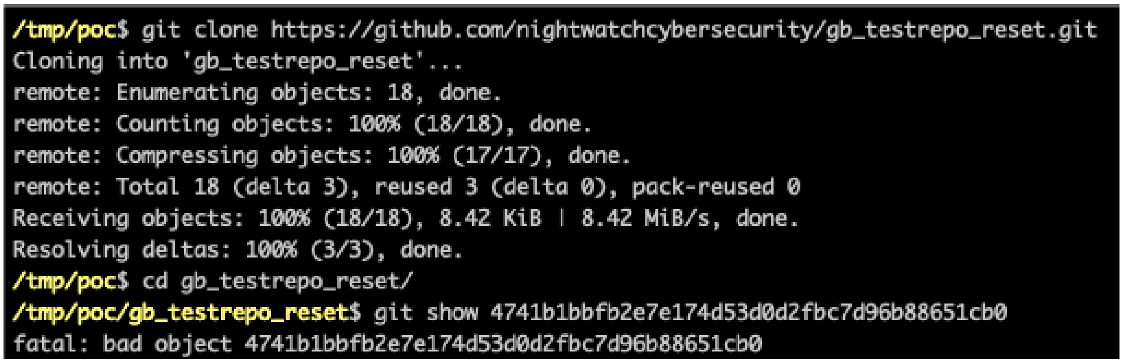
- gb_testrepo_delete (https://github.com/nightwatchcybersecurity/gb_testrepo_delete) – secret is retained after a branch is deleted
- gb_testrepo_reset (https://github.com/nightwatchcybersecurity/gb_testrepo_reset) – secret is retained after the git history is reset

If you try to clone the repository without the "–mirror" option, and retrieve the secret, it will not work:



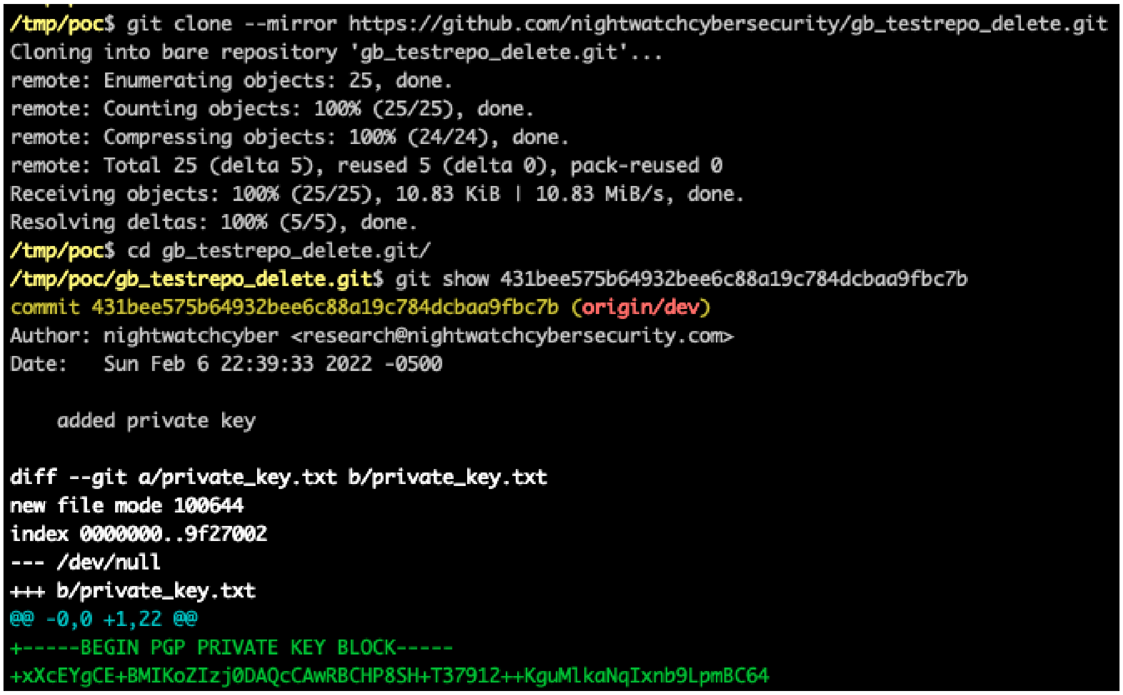
```
/tmp/poc$ git clone https://github.com/nightwatchcybersecurity/gb_testrepo_delete.git
Cloning into 'gb_testrepo_delete'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 22 (delta 4), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (22/22), 9.67 KiB | 9.67 MiB/s, done.
Resolving deltas: 100% (4/4), done.
/tmp/poc$ cd gb_testrepo_delete/
/tmp/poc/gb_testrepo_delete$ git show 431bee575b64932bee6c88a19c784dcbaa9fbc7b
fatal: bad object 431bee575b64932bee6c88a19c784dcbaa9fbc7b
```

And:



```
/tmp/poc$ git clone https://github.com/nightwatchcybersecurity/gb_testrepo_reset.git
Cloning into 'gb_testrepo_reset'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 18 (delta 3), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (18/18), 8.42 KiB | 8.42 MiB/s, done.
Resolving deltas: 100% (3/3), done.
/tmp/poc$ cd gb_testrepo_reset/
/tmp/poc/gb_testrepo_reset$ git show 4741b1bbfb2e7e174d53d0d2fbc7d96b88651cb0
fatal: bad object 4741b1bbfb2e7e174d53d0d2fbc7d96b88651cb0
```

If you try the same with the "–mirror" option, you can now retrieve the secret (also note the larger number of objects retrieved):

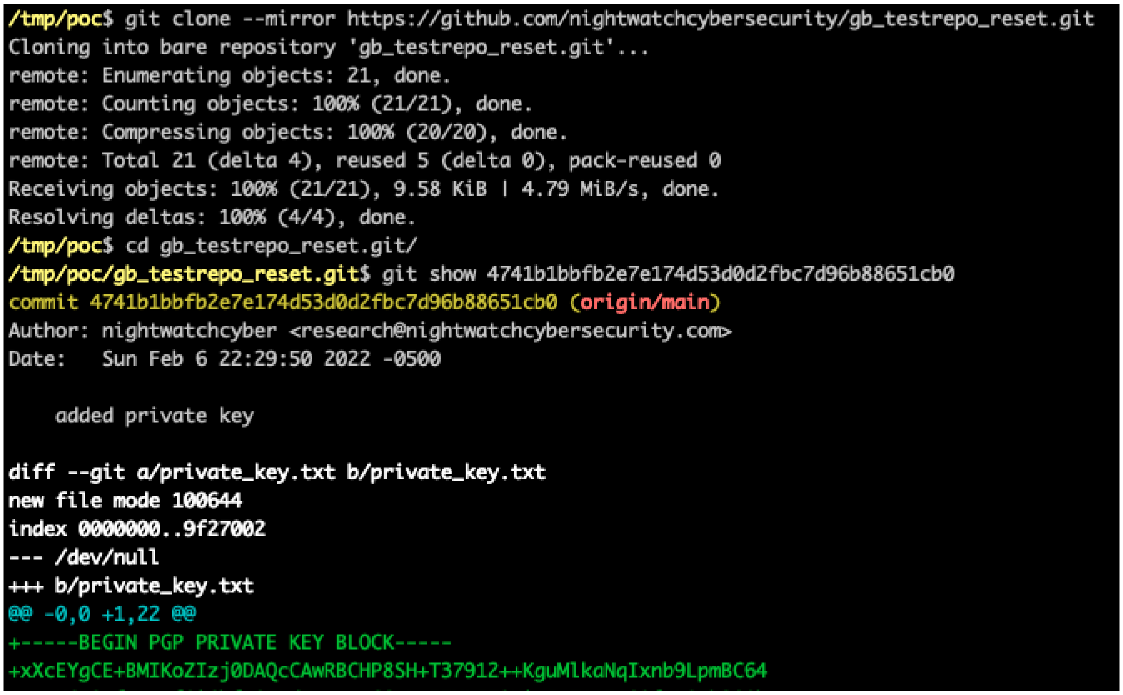


```
/tmp/poc$ git clone --mirror https://github.com/nightwatchcybersecurity/gb_testrepo_delete.git
Cloning into bare repository 'gb_testrepo_delete.git'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 25 (delta 5), reused 5 (delta 0), pack-reused 0
Receiving objects: 100% (25/25), 10.83 KiB | 10.83 MiB/s, done.
Resolving deltas: 100% (5/5), done.
/tmp/poc$ cd gb_testrepo_delete.git/
/tmp/poc/gb_testrepo_delete.git$ git show 431bee575b64932bee6c88a19c784dcbaa9fbc7b
commit 431bee575b64932bee6c88a19c784dcbaa9fbc7b (origin/dev)
Author: nightwatchcyber <research@nightwatchcybersecurity.com>
Date:   Sun Feb 6 22:39:33 2022 -0500

    added private key

diff --git a/private_key.txt b/private_key.txt
new file mode 100644
index 0000000..9f27002
--- /dev/null
+++ b/private_key.txt
@@ -0,0 +1,22 @@
+-----BEGIN PGP PRIVATE KEY BLOCK-----
+xXcEYgCE+BMIKoZIzj0DAQcCAwRBCHP8SH+T37912++KguMlkaNqIxnb9LpmBC64
```

And:

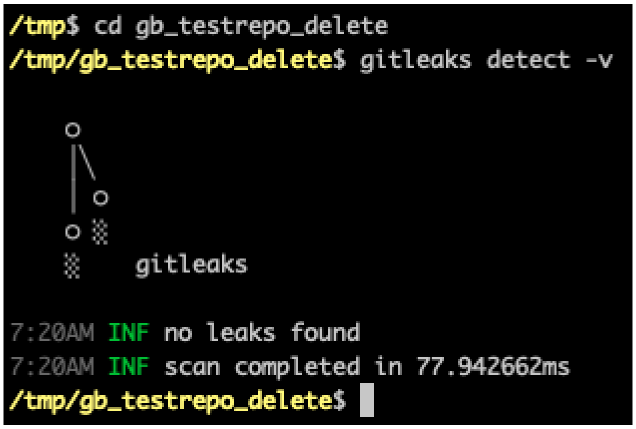


```
/tmp/poc$ git clone --mirror https://github.com/nightwatchcybersecurity/gb_testrepo_reset.git
Cloning into bare repository 'gb_testrepo_reset.git'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 21 (delta 4), reused 5 (delta 0), pack-reused 0
Receiving objects: 100% (21/21), 9.58 KiB | 4.79 MiB/s, done.
Resolving deltas: 100% (4/4), done.
/tmp/poc$ cd gb_testrepo_reset.git/
/tmp/poc/gb_testrepo_reset.git$ git show 4741b1bbfb2e7e174d53d0d2fbc7d96b88651cb0
commit 4741b1bbfb2e7e174d53d0d2fbc7d96b88651cb0 (origin/main)
Author: nightwatchcyber <research@nightwatchcybersecurity.com>
Date:   Sun Feb 6 22:29:50 2022 -0500

    added private key

diff --git a/private_key.txt b/private_key.txt
new file mode 100644
index 0000000..9f27002
--- /dev/null
+++ b/private_key.txt
@@ -0,0 +1,22 @@
+-----BEGIN PGP PRIVATE KEY BLOCK-----
+xXcEYgCE+BMIKoZIzj0DAQcCAwRBCHP8SH+T37912++KguMlkaNqIxnb9LpmBC64
```

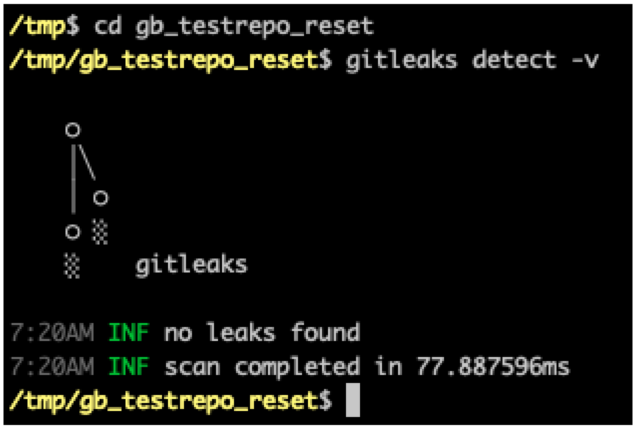
If you run gitleaks (https://github.com/zricethezav/gitleaks) on the cloned repositories, no secrets are found:



```
/tmp$ cd gb_testrepo_delete
/tmp/gb_testrepo_delete$ gitleaks detect -v

    ○
    │╲
    │ ○
    ○ ░
    ░    gitleaks

7:20AM INF no leaks found
7:20AM INF scan completed in 77.942662ms
/tmp/gb_testrepo_delete$ 
```

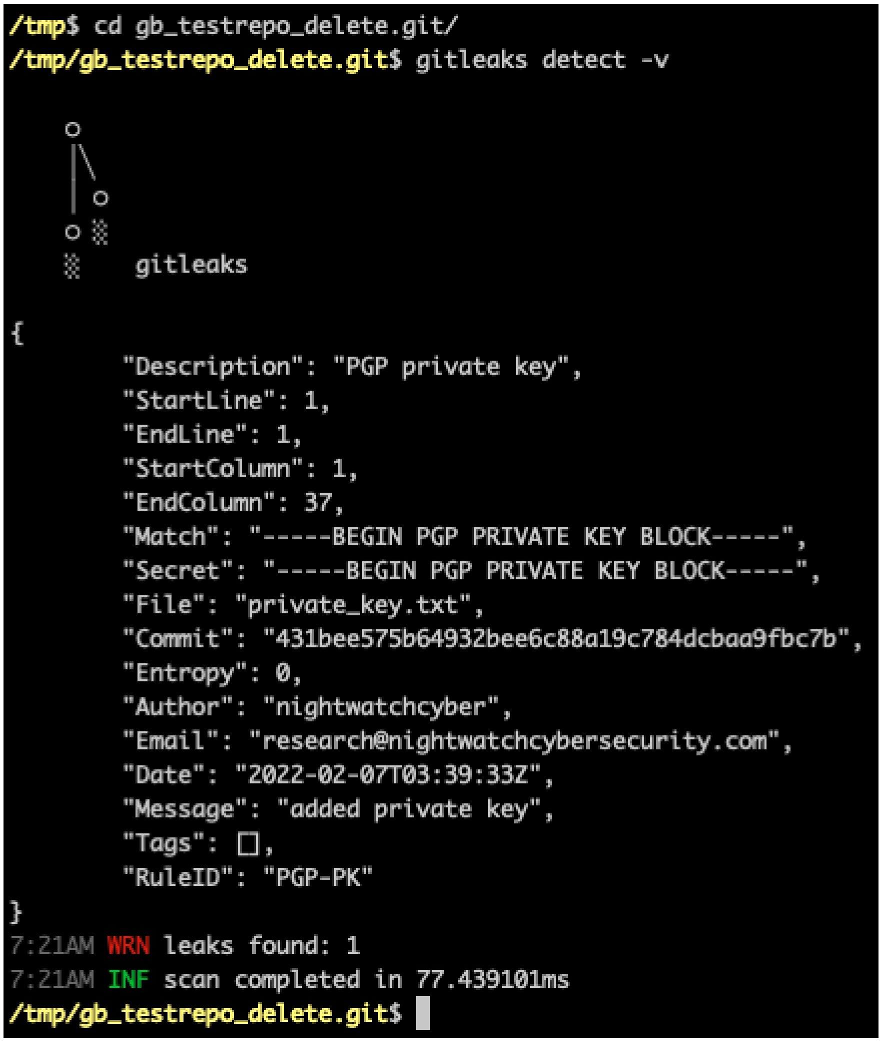


```
/tmp$ cd gb_testrepo_reset
/tmp/gb_testrepo_reset$ gitleaks detect -v

    ○
    │╲
    │ ○
    ○ ░
    ░    gitleaks

7:20AM INF no leaks found
7:20AM INF scan completed in 77.887596ms
/tmp/gb_testrepo_reset$ 
```

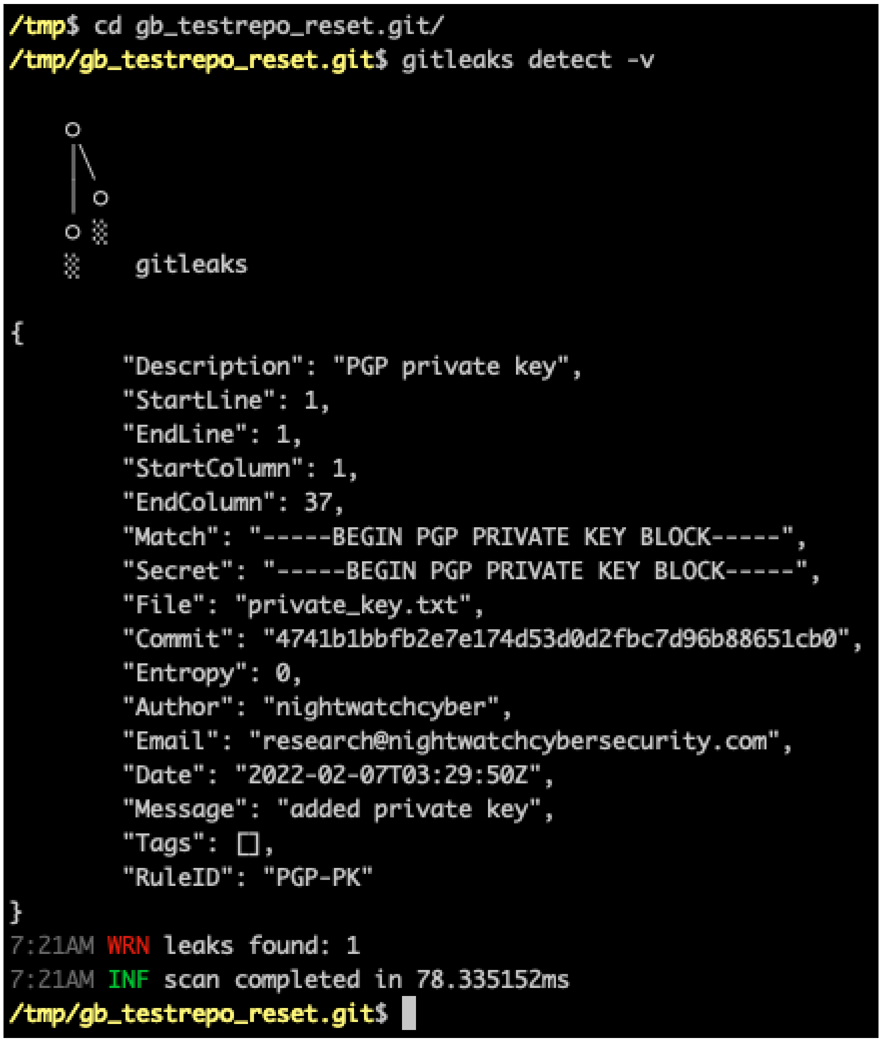
However, running gitleaks (https://github.com/zricethezav/gitleaks) on the mirrored copies, finds the secrets stashed in deleted areas:



```
/tmp$ cd gb_testrepo_delete.git/
/tmp/gb_testrepo_delete.git$ gitleaks detect -v

    ○
    │╲
    │ ○
    ○ ░
    ░    gitleaks

{
        "Description": "PGP private key",
        "StartLine": 1,
        "EndLine": 1,
        "StartColumn": 1,
        "EndColumn": 37,
        "Match": "-----BEGIN PGP PRIVATE KEY BLOCK-----",
        "Secret": "-----BEGIN PGP PRIVATE KEY BLOCK-----",
        "File": "private_key.txt",
        "Commit": "431bee575b64932bee6c88a19c784dcbaa9fbc7b",
        "Entropy": 0,
        "Author": "nightwatchcyber",
        "Email": "research@nightwatchcybersecurity.com",
        "Date": "2022-02-07T03:39:33Z",
        "Message": "added private key",
        "Tags": [],
        "RuleID": "PGP-PK"
}
7:21AM WRN leaks found: 1
7:21AM INF scan completed in 77.439101ms
/tmp/gb_testrepo_delete.git$ 
```



```
/tmp$ cd gb_testrepo_reset.git/
/tmp/gb_testrepo_reset.git$ gitleaks detect -v

    ○
    │╲
    │ ○
    ○ ░
    ░    gitleaks

{
        "Description": "PGP private key",
        "StartLine": 1,
        "EndLine": 1,
        "StartColumn": 1,
        "EndColumn": 37,
        "Match": "-----BEGIN PGP PRIVATE KEY BLOCK-----",
        "Secret": "-----BEGIN PGP PRIVATE KEY BLOCK-----",
        "File": "private_key.txt",
        "Commit": "4741b1bbfb2e7e174d53d0d2fbc7d96b88651cb0",
        "Entropy": 0,
        "Author": "nightwatchcyber",
        "Email": "research@nightwatchcybersecurity.com",
        "Date": "2022-02-07T03:29:50Z",
        "Message": "added private key",
        "Tags": [],
        "RuleID": "PGP-PK"
}
7:21AM WRN leaks found: 1
7:21AM INF scan completed in 78.335152ms
/tmp/gb_testrepo_reset.git$ 
```

# Tooling

There are plenty of existing tools out there that can manipulate git repositories, scan them for secrets and remove specific commits. During our research, we used git for checking out repositories, git-filter-repo (https://github.com/newren/git-filter-repo) for figuring out the delta between cloned and mirrored copies of the same repository, and gitleaks (https://github.com/zricethezav/gitleaks) to scan for secrets.

For examples on how to use these tools, please see sample scripts that we have published to GitHub (https://github.com/nightwatchcybersecurity/gitbleed_tools).

# Mitigations

Organizations can mitigate this by analyzing a larger part of their repositories using the "–mirror" option and remove sensitive data using tools like BFG (https://rtyley.github.io/bfg-repo-cleaner/) or git-filter-repo (https://github.com/newren/git-filter-repo). Garbage collection and pruning in git is also recommended.

Organizations should not analyze regular cloned copies (without the "–mirror" option) since that may provide a false sense of security, and should not rely on methods of removing secrets such as deleting a branch or rewinding history via the "git reset (https://git-scm.com/docs/git-reset)" command.

## Branding

(https://wwwsnightwatchcybersecuritycom.files.wordpress.com/2022/02/gitbleed_icon-3.png)

There seems to be a recent trend to name vulnerabilities. While we think it's silly, why not go with the flow?

Therefore we named this one "**GitBleed**", since it leads to bleeding of secrets from repositories – with a mirrored logo (https://github.com/nightwatchcybersecurity/gitbleed_tools/blob/main/gitbleed_icon.png) since it involves mirrored repositories.

## Changelog

2022-02-11: Initial publication

2022-02-26: Added CVE reference

This site uses Akismet to reduce spam. Learn how your comment data is processed (https://akismet.com/privacy/).