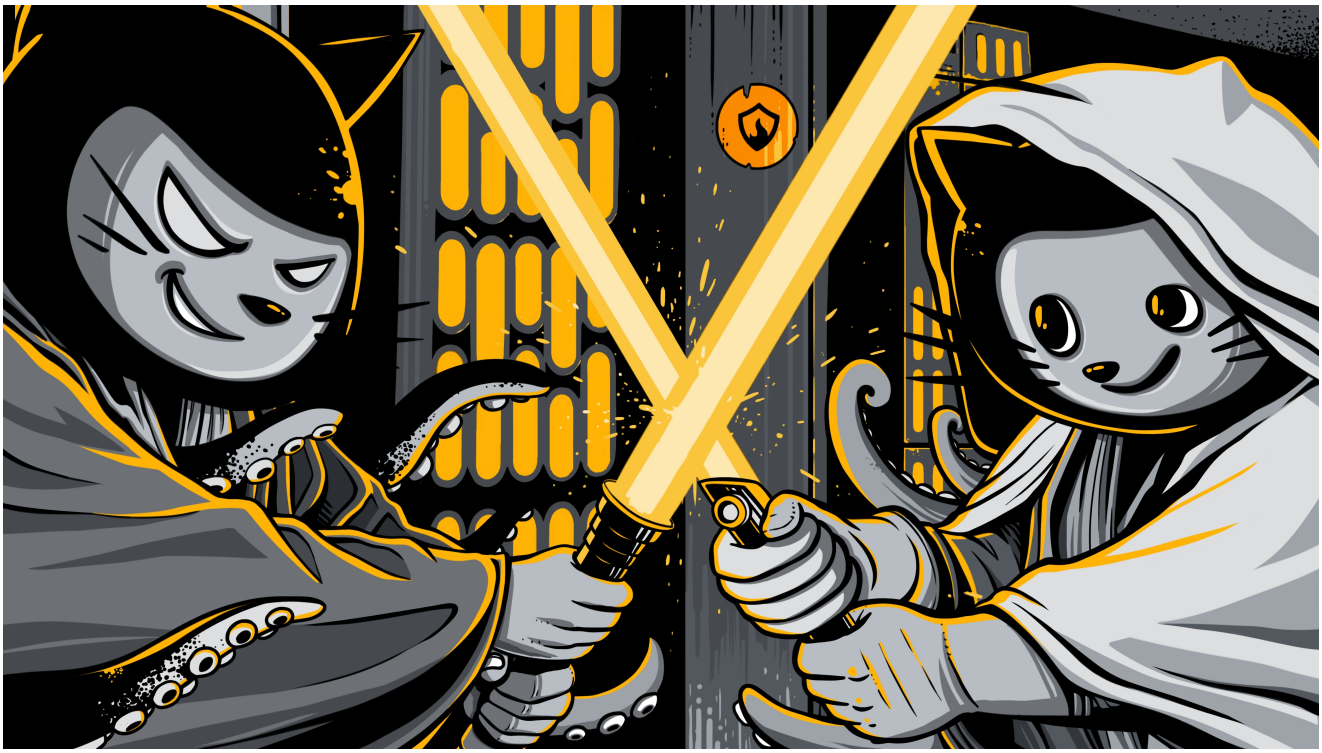
Attack of the clones: Git clients remote code execution

blog.blazeinfosec.com/attack-of-the-clones-github-desktop-remote-code-execution/

Blaze Information Security November 5, 2020



Introduction

This post is a rather unusual story of a vulnerability that could be leveraged as a supply chain attack and used to attack millions of software developers around the world. It is also a tale of a bug collision that paid a bounty to one reporter and assigned the CVE to another!

The main focus of this blog post is GitHub Desktop. Other Git clients such as GitKraken, Git-Tower and SourceTree were also found to be vulnerable, however these have different exploitation scenarios that require user interaction.

Brief description of the issue

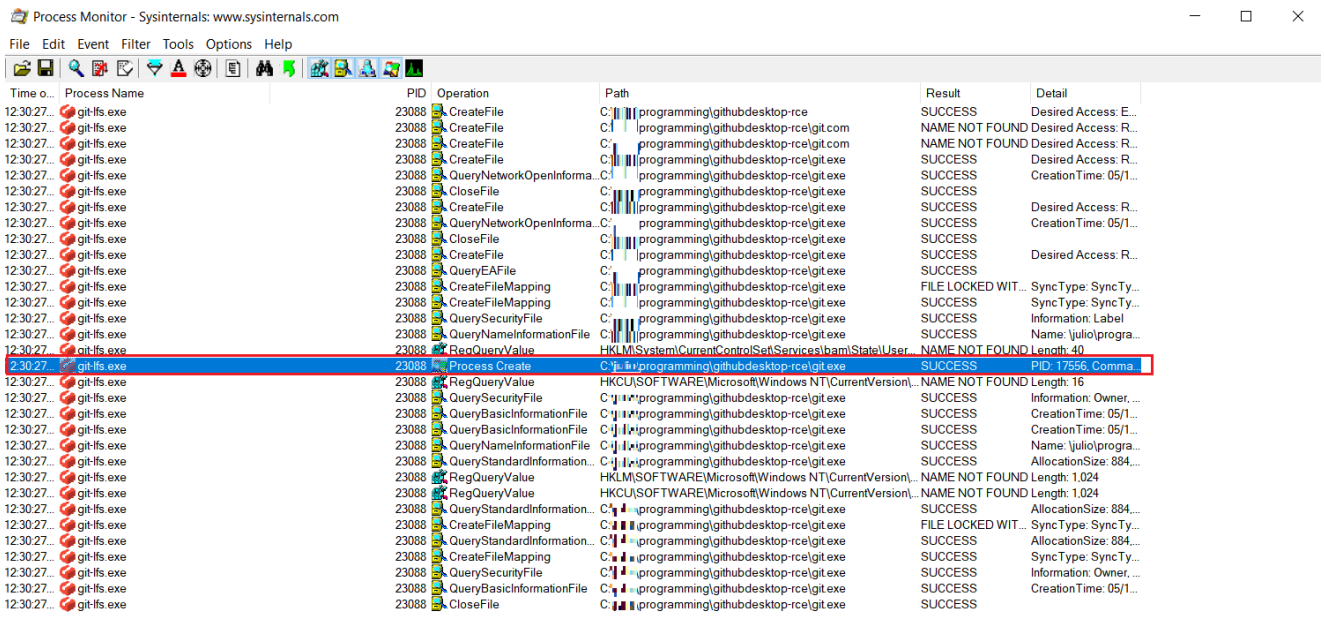
As part of GitHub Desktop's default repository cloning process, among other actions it calls the executable git-lfs.

From git-lfs's official page "Git Large File Storage (LFS) replaces large files such as audio samples, videos, datasets, and graphics with text pointers inside Git, while storing the file contents on a remote server like GitHub.com or GitHub Enterprise."

git-lfs is then called on the current cloned repository, already present in disk.

A vulnerability was discovered when cloning a repository with a especially crafted file in the root directory. Upon cloning such malicious repository, code execution is achieved with the same privileges as the affected user running GitHub Desktop.

When inspecting the desktop application under Proccess Monitor, it was noticed several actions of git-lfs called a git executable from inside the newly cloned repository.



By placing a malicious git in the root of the malicious repo, a user that clones it will have the malicious git executed by git-lfs, all of this behind the scenes and transparent to the user.

The cloning process happens normally and there is no visual indication that a malicious binary was ran instead of the original git executable.

Root cause analysis

We attempted to understand the root cause of this vulnerability by reading git-lfs's code. The authors of this write-up tried to put together as much information as possible in order to understand the cause of the problem.

From https://github.com/git-lfs/git-lfs/blob/master/commands/command_clone.go

From line 23:

```
func cloneCommand(cmd *cobra.Command, args []string) {
        requireGitVersion()

        if git.IsGitVersionAtLeast("2.15.0") {
                msg := []string{
                        "WARNING: 'git lfs clone' is deprecated and will not be
updated",
                        "          with new flags from 'git clone'",
                        "",
                        "'git clone' has been updated in upstream Git to have
comparable",
                        "speeds to 'git lfs clone'.",
                }

                fmt.Fprintln(os.Stderr, strings.Join(msg, "\n"))
        }
```

The function above seems to be called when a clone action takes place. Right in the beginning the function calls requireGitVersion() and then checks whether the version is at least 2.15.0.

From https://github.com/git-lfs/git-lfs/blob/master/commands/commands.go

Line 537:

```
func requireGitVersion() {
        minimumGit := "1.8.2"

        if !git.IsGitVersionAtLeast(minimumGit) {
                gitver, err := git.Version()
                if err != nil {
                        Exit("Error getting git version: %s", err)
                }
                Exit("git version >= %s is required for Git LFS, your version: %s",
minimumGit, gitver)
        }
}
```

It calls !git.IsGitVersionAtLeast(minimumGit) which can be found in https://github.com/git-lfs/git-lfs/blob/master/git/version.go

From line 28:

```
func IsGitVersionAtLeast(ver string) bool {
        gitver, err := Version()
        if err != nil {
                tracerx.Printf("Error getting git version: %v", err)
                return false
        }
        return IsVersionAtLeast(gitver, ver)
}
```

Version() can be found in the same file version.go on line 18:

```
func Version() (string, error) {
        gitVersionOnce.Do(func() {
                gitVersion, gitVersionErr =
                        subprocess.SimpleExec("git", "version")
        })
        return gitVersion, gitVersionErr
}
```

The function calling chain is the following:
cloneCommand() -> requireGitVersion() -> IsGitVersionAtLeast() -> Version()

We can see that in line 21 in version.go, there is inside the function Version() a call to subprocess.SimpleExec() - it executes the git binary with the argument "version".

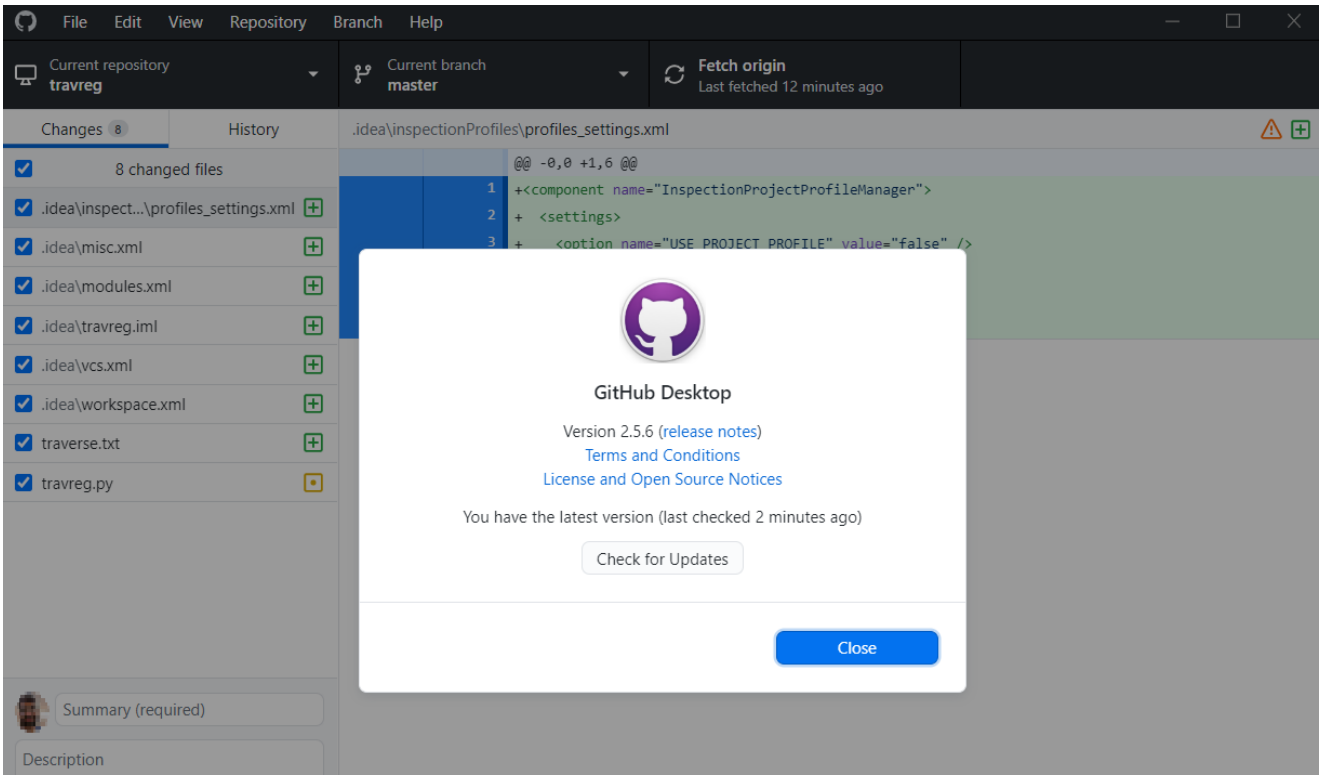
There is an implicit assumption from git-lfs that by executing an external binary, the operating system will first look it up from the system's PATH environment variable. This is the true root cause of the vulnerability.

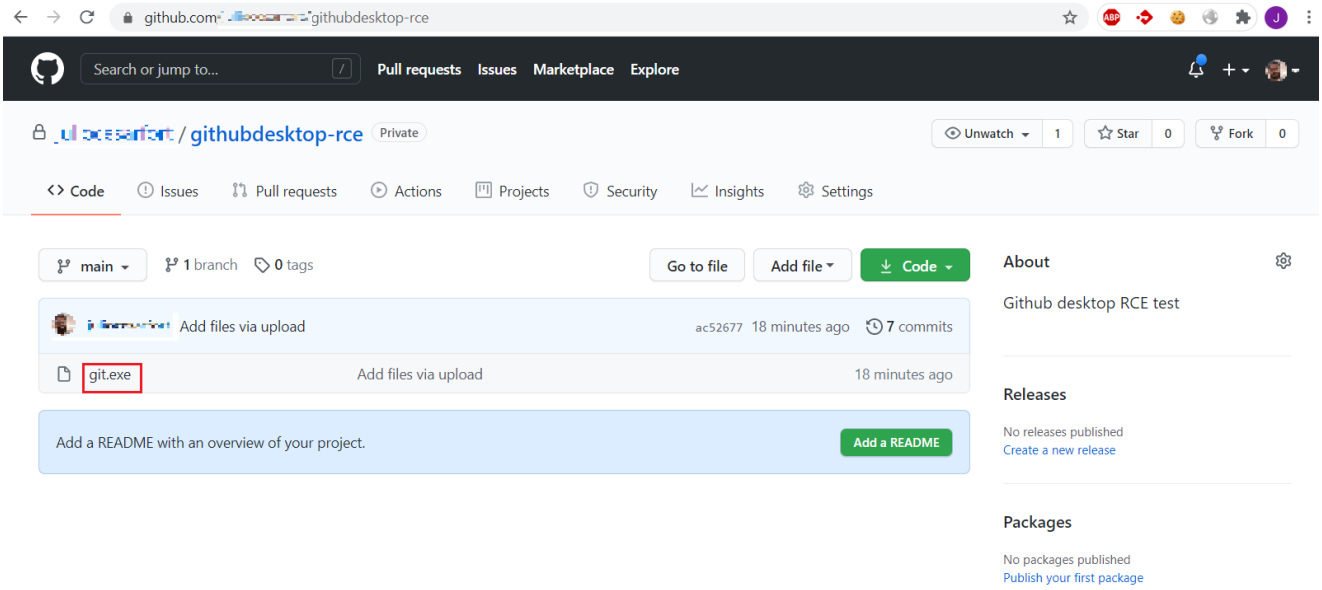
Calling the subprocess 'git' works as intended in GitHub Desktop for MacOS. In Windows, however, the search order favors the current directory first and only then it searches in the order specified in PATH.

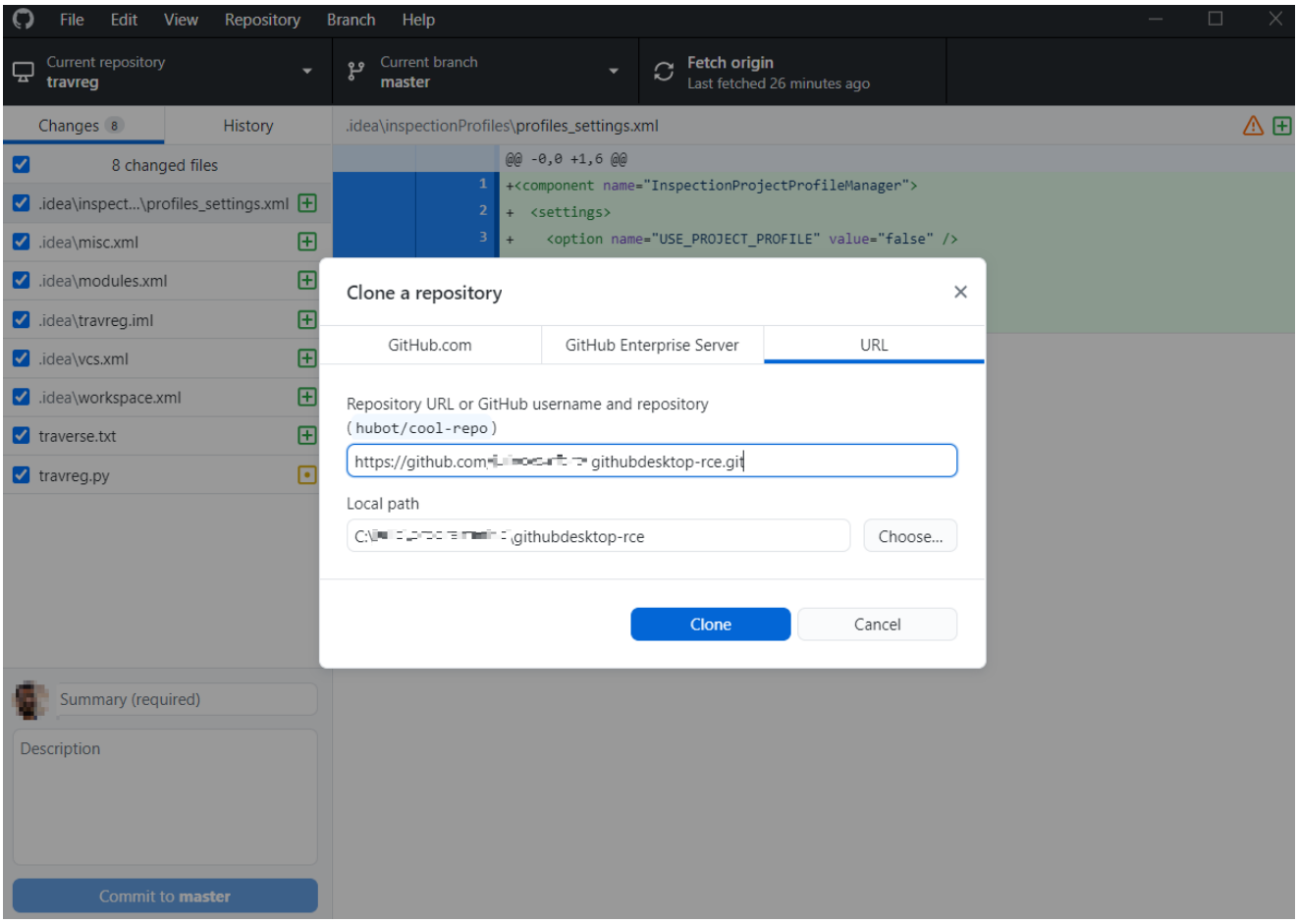
Exploitation

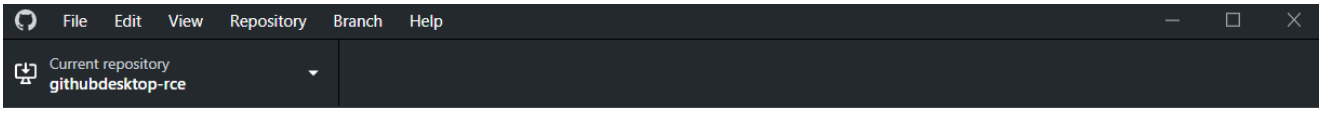
By copying calc.exe, renaming it to git.exe and committing/pushing it into the root of the repository, the attacker sets up the crafted repository that will be used to later exploit users.

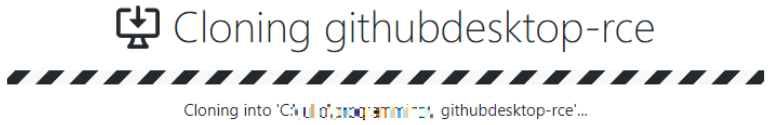
When a developer clones this repository using GitHub Desktop for Windows, its subcomponent git-lfs will work on the cloned folder as its current directory, meaning all subsequent calls to the binary 'git' will be instead calling the malicious git.exe not the one present in the PATH.

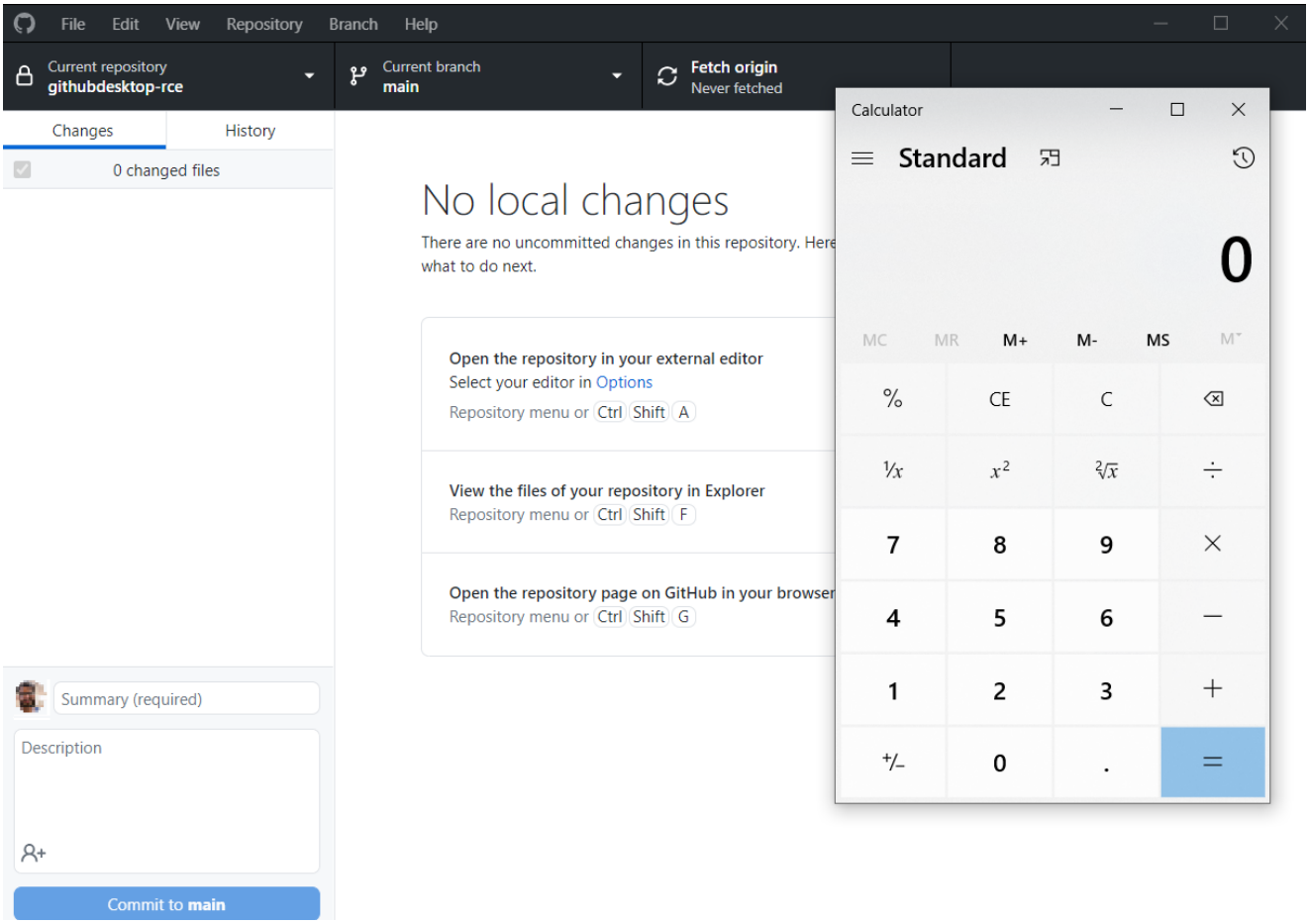

File Edit View Repository Branch Help
Current repository
travreg
Current branch
master
Fetch origin
Last fetched 12 minutes ago
Changes 8
History
8 changed files
.idea\inspect...\profiles_settings.xml
.idea\misc.xml
.idea\modules.xml
.idea\travreg.iml
.idea\vcs.xml
.idea\workspace.xml
traverse.txt
travreg.py
.idea\inspectionProfiles\profiles_settings.xml
@@ -0,0 +1,6 @@
+<component name="InspectionProjectProfileManager">
+ <settings>
GitHub Desktop
Version 2.5.6 (release notes)
Terms and Conditions
License and Open Source Notices
You have the latest version (last checked 2 minutes ago)
Check for Updates
Close
Summary (required)
Description


github.com/githubdesktop-rce
Search or jump to...
Pull requests
Issues
Marketplace
Explore
/ githubdesktop-rce
Private
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
Add files via upload
ac52677 18 minutes ago
7 commits
git.exe
Add files via upload
18 minutes ago
Add a README with an overview of your project.
Add a README
About
Github desktop RCE test
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package


File Edit View Repository Branch Help
Current repository
travreg
Current branch
master
Fetch origin
Last fetched 26 minutes ago
Changes 8
History
8 changed files
.idea\inspect...\profiles_settings.xml
.idea\misc.xml
.idea\modules.xml
.idea\travreg.iml
.idea\vcs.xml
.idea\workspace.xml
traverse.txt
travreg.py
.idea\inspectionProfiles\profiles_settings.xml
@@ -0,0 +1,6 @@
+<component name="InspectionProjectProfileManager">
+ <settings>
+ <option name="USE_PROJECT_PROFILE" value="false" />
Clone a repository
GitHub.com
GitHub Enterprise Server
URL
Repository URL or GitHub username and repository
(hubot/cool-repo)
githubdesktop-rce.git
Local path
githubdesktop-rce
Choose...
Clone
Cancel
Summary (required)
Description
Commit to master


File Edit View Repository Branch Help
Current repository
githubdesktop-rce


Cloning githubdesktop-rce
Cloning into 'C githubdesktop-rce'...



Below is a the video of exploiting this issue on GitHub Desktop for Windows:

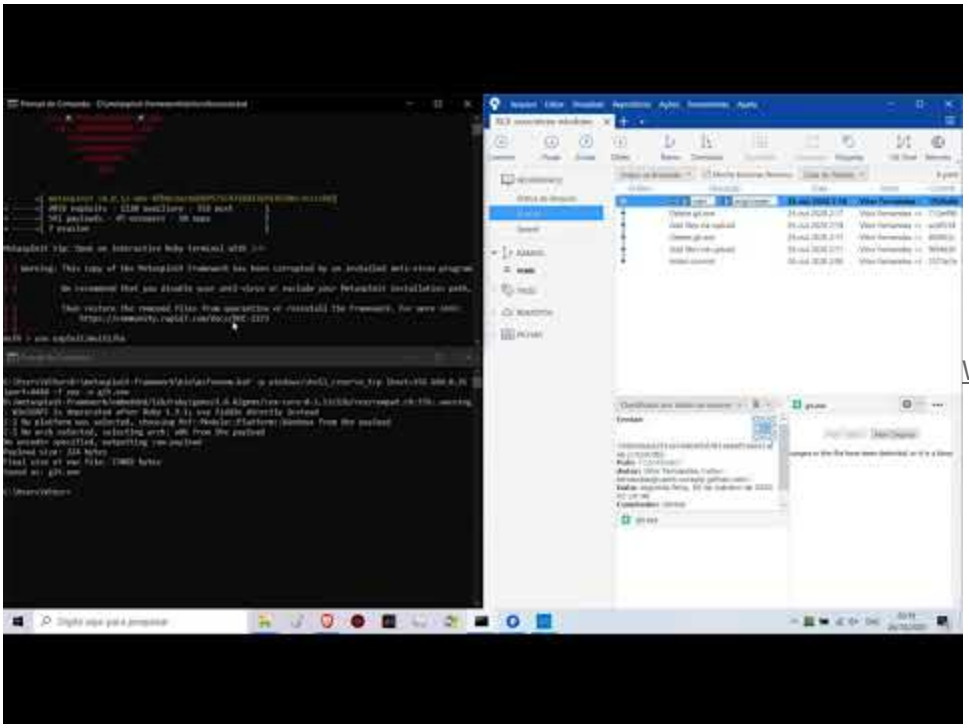
Watch Video At:

https://youtu.be/p8tbHNBbO1U

Other clients

During our research we noticed SourceTree

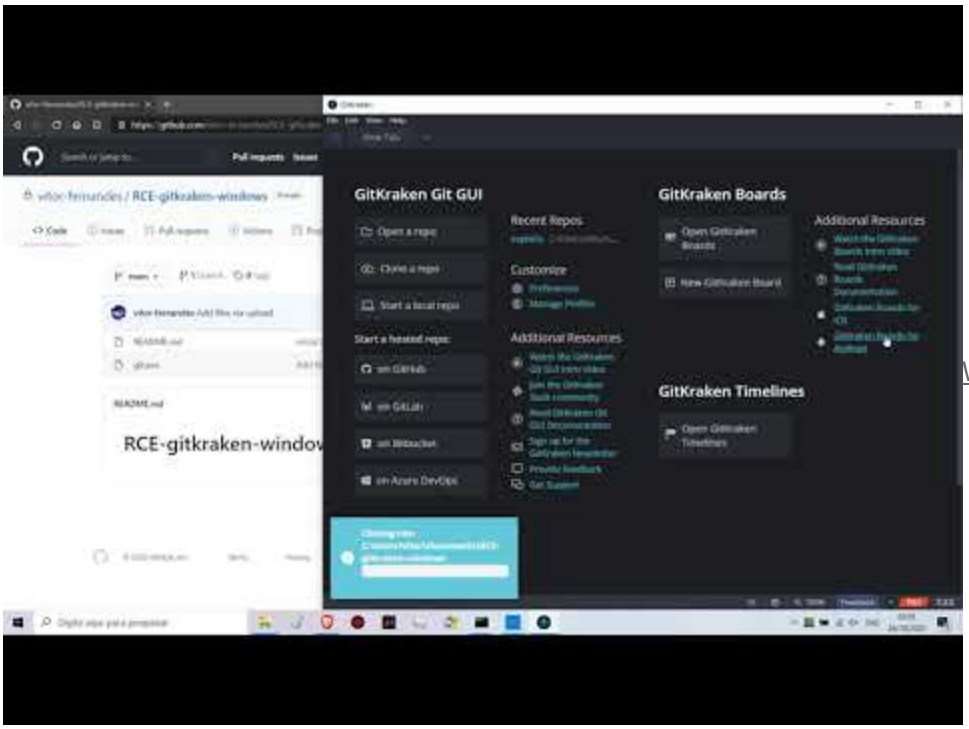
SourceTree



Watch Video At:

https://youtu.be/zOXUdP50VYM

GitKraken



Watch Video At:

https://youtu.be/RmLCjgc0ppo

Credits

The vulnerability was simultaneously discovered and researched by Vitor Fernandes and Julio Fort of Blaze Information Security. The independent researcher Dawid Golunski (dobra robota, amigo!) apparently discovered the issue a few days earlier and submitted it to MITRE, while Vitor submitted it to GitHub's bug bounty.

Vitor collected the bounty from GitHub's program on HackerOne as his report arrived earlier and Dawid was assigned the CVE.

References

[1] https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/path]
[2] https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc753427(v=ws.11)
[3] https://github.com/git-lfs/git-lfs/security/advisories/GHSA-4g4p-42wc-9f3m (Collision with Dawid Golunski)