

New issue

[Jump to bottom](#)

uri module set string with masked content into content and json output #68400

🔒 Closed hungluong5791 opened this issue on Mar 23, 2020 · 12 comments · Fixed by [#69653](#)

Labels affects_2.9 **bug** **easyfix** module net_tools **P3** **support:core**

hungluong5791 commented on Mar 23, 2020

SUMMARY

uri module set string with masked content into content and json output

ISSUE TYPE

- Bug Report

COMPONENT NAME

uri

ANSIBLE VERSION

```
ansible 2.9.0
config file = None
configured module search path = ['/Users/hungluong/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
ansible python module location = /Users/hungluong/Library/Python/3.7/lib/python/site-packages/ansible
executable location = /Users/hungluong/Library/Python/3.7/bin/ansible
python version = 3.7.6 (default, Dec 30 2019, 19:38:26) [Clang 11.0.0 (clang-1100.0.33.16)]
```

CONFIGURATION

N/A

OS / ENVIRONMENT

STEPS TO REPRODUCE

```
- hosts: localhost
  connection: local
  tasks:
    - name: send request
      uri:
        url: "https://postman-echo.com/get?name=something-with-admin"
        user: admin
        password: admin
        method: GET
        force_basic_auth: yes
        return_content: yes
        status_code: 200
        register: response

    - name: extract value
      vars:
        query: args.name
      set_fact:
        value_content: "{{ response.content }}"
        value_content_parsed: "{{ response.content | from_json | json_query(query) }}"
        value_json: "{{ response.json.args.name }}"

    - name: debug
      debug:
        msg:
          - "{{ 'something-with-admin' in value_json }}"
          - "{{ 'something-with-admin' in value_content }}"
          - "{{ 'something-with-admin' in value_content_parsed }}"
          - "{{ 'something-with-*****' in value_json }}"
          - "{{ 'something-with-*****' in value_content }}"
          - "{{ 'something-with-*****' in value_content_parsed }}"
```

EXPECTED RESULTS

The module should return the json/content value with the correct values

ACTUAL RESULTS

The module seems to apply sensitive info masking (*****) to value matching username/password in its output

```
"msg": [
  false,
  false,
  false,
  true,
  false,
  true
]
```

ansibot commented on Mar 23, 2020

Contributor

Files identified in the description:

- [lib/ansible/modules/net_tools/basics/uri.py](#)

If these files are inaccurate, please update the `component name` section of the description or use the `!component bot` command.

[click here for bot help](#)

bcoca added P3 and removed needs_triage labels on Apr 14, 2020

bcoca commented on Apr 14, 2020

Member

Actually we should be masking the non json keys also.

bcoca added the easyfix label on Apr 14, 2020

ansibot commented on May 15, 2020

Contributor

Files identified in the description:

- [lib/ansible/modules/uri.py](#)

If these files are incorrect, please update the `component name` section of the description or use the `!component bot` command.

[click here for bot help](#)

rubber-side-down commented on May 20, 2020

Actually we should be masking the non json keys also.

It's masking the non-json (`response.content`) value as well. Adding the facts to the debug msg confirms this:

```
- hosts: localhost
  connection: local
  tasks:
    - name: send request
      uri:
        url: "https://postman-echo.com/get?name=something-with-admin"
        user: admin
        password: admin
        method: GET
        force_basic_auth: yes
        return_content: yes
        status_code: 200
        register: response

    - name: extract value
      vars:
        query: args.name
      set_fact:
        value_json: "{{ response.json.args.name }}"
        value_content: "{{ response.content }}"
        value_content_parsed: "{{ response.content | from_json | json_query (query) }}"

    - name: debug
      debug:
        msg:
          - "{{ value_json }}"
          - "{{ value_content }}"
          - "{{ value_content_parsed }}"
          - "{{ 'something-with-admin' in value_json }}"
          - "{{ 'something-with-admin' in value_content }}"
          - "{{ 'something-with-admin' in value_content_parsed }}"
          - "{{ 'something-with-admin' in value_json }}"
          - "{{ 'something-with-*****' in value_json }}"
          - "{{ 'something-with-*****' in value_content }}"
          - "{{ 'something-with-*****' in value_content_parsed }}"
```

```
ok: [localhost] => {
  "msg": [
    "something-with-*****",
    {
      "args": {
        "name": "something-with-*****"
      },
      "headers": {
        "accept-encoding": "identity",
        "authorization": "Basic YWRtaW46YWRtaW4=",
        "host": "postman-echo.com",
        "user-agent": "ansible-httpget",
        "x-amzn-trace-id": "Root=1-Sec4de30-6007601b0412140ca002d5eb",
        "x-forwarded-port": "443",
        "x-forwarded-proto": "https"
      },
      "url": "https://postman-echo.com/get?name=something-with-*****"
    },
    "something-with-*****",
    false,
    false,
    false,
    true,
    false,
    true
  ]
}
```

'something-with-*****' in value_content returns False because the test is incorrect. It tests whether something-with-***** is a key in value_content rather than whether the string exists in the content. Changing the test to 'something-with-*****' in value_content.__str__() returns True .

@bcoca, is there work to do here? It sounds like the request is to stop masking the values, but your comment indicates that these values should be masked.

hungluong5791 commented on May 20, 2020

Author

It tests whether something-with-***** is a key in value_content rather than whether the string exists in the content

@rubber-side-down Why is that the case? The module example seems to indicate that it is just a string:

```
- name: Check that a page returns a status 200 and fail if the word AWESOME is not in the page contents
  uri:
    url: http://www.example.com
    return_content: yes
  register: this
  failed_when: "'AWESOME' not in this.content"
```

rubber-side-down commented on May 21, 2020

@hungluong5791 I'm uncertain why this is the case, but it's pretty clear from modifying the debug msg spec that this is the case.

My best guess is that the response.content is being decoded automatically based on the content-type header. The content-type header is "text/html; charset=UTF-8" for <http://www.example.com>, but it is "application/json; charset=utf-8" for <https://postman-echo.com/get?name=something-with-admin>.

sivel commented on May 21, 2020

Member

@bcoca, is there work to do here? It sounds like the request is to stop masking the values, but your comment indicates that these values should be masked.

Yes, the request to not mask the values is invalid. In fact this issues identified that we need to mask more data.

sivel added a commit to sivel/ansible that referenced this issue on May 21, 2020

Strip no log values from module response keys. Fixes [ansible#68400](#)

2c0ac03

sivel mentioned this issue on May 21, 2020

Strip no log values from module response keys #69653

↳ Merged

samdoran closed this as completed in [#69653](#) on May 21, 2020

samdoran pushed a commit that referenced this issue on May 21, 2020

Strip no log values from module response keys. Fixes [#68400](#) ([#69653](#))

✗ e0f25a2

hungluong5791 commented on May 21, 2020 • edited

Author

@samdoran @sivel Hi, I'm a bit confused. Why is the data need to be masked? They are logically irrelevant from each other, and this actually blocks our API integration workflow.

hungluong5791 commented on May 21, 2020

Author

@rubber-side-down I see, that might be it. But do you think that behaviour is inherently confusing and unnecessary? That's the entire purpose of the response's `json` key, no?

rubber-side-down commented on May 22, 2020

@hungluong5791, I could go either way on the behavior of request content marshaling/deserialization, but it's beyond the scope of this issue.

sivel commented on Jun 1, 2020

Member

@hungluong5791 We purposefully mask return values to hide no log values. Because you have chosen a password that also appears in the output, the values are masked. You pointed out a security flaw in our code, and as such, we addressed it.

Consider changing your password to some value that is not included in the return values from the API.

hungluong5791 commented on Jun 2, 2020

Author

@sivel Thank you for getting back. Yes I understand the behaviour and I think I understand where it is coming from. But please consider the following points:

- First, my specific use case:
 - we are provisioning an off-the-shelf system and having to create users & groups in that system through the API.
 - The default credentials is `admin / admin` (which cannot be changed, only disabled)
 - We have to create admin groups belonging to departments in the form of `group1-admin`, `group2-admin` etc., checking first to see if they exists for idempotency
 - The API does not support directly querying if a group exists, instead only provides a fuzzy search API - searching `admin` yields `admin1`, `admin2` and such.
--> thus the need of looking for values containing the credentials (`admin`) in the response, which currently blocked.
- Secondly, `uri` is a generic API-calling module - as such it should not "police" external API behaviours. If an API decide to return the credentials in its response (like say `httpbin`), it is up to the user to decide if they should use that API, not the module. Not logging the value to output is perfectly sensible, but the user should still be able to use the data. Aggressively managing and requiring users to change their data just to use the module is backward, imo.
- Thirdly, while I haven't tried it, I'd presume I could achieve what I want by using `shell` and `curl` (or just `curl` in a shell script) - which is neither portable, convenient nor any more secure. Pardon if I am ignorant about all the security implications from this, but if I could already do it, shouldn't Ansible make it easier, not blocking it?

Thanks

Assignees

No one assigned

Labels

affects_2.9 **bug** **easyfix** module net_tools **P3** **support:core**

Projects


None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

 **Strip no log values from module response keys**
sivel/ansible

5 participants

