two random password lookups in same task return same value #34144

Closed richlv opened this issue on Dec 21, 2017 · 19 comments · Fixed by #67429

Labels affects_2.3 bug has_pr support:core test

richlv commented on Dec 21, 2017 • edited Contributor

ISSUE TYPE

- Bug Report

COMPONENT NAME

lookups

ANSIBLE VERSION

```
2.3.0.0, 2.4.1.0
```

CONFIGURATION

no custom config

OS / ENVIRONMENT

Linux

SUMMARY

two consecutive facts from password lookup with same length get identical values.
separate tasks or different length result in different values, as expected.

STEPS TO REPRODUCE

```
- name: Set random passwords for clusters
  set_fact:
    password1: "{{ lookup('password', '/dev/null length=20') }}"
    password2: "{{ lookup('password', '/dev/null length=20') }}"
  delegate_to: localhost
  delegate_facts: True
- debug:
    msg: "{{hostvars['localhost'].password1}} {{hostvars['localhost'].password2}}"
```

EXPECTED RESULTS

two different values

ACTUAL RESULTS

two identical values

changing length for one of these, or using separate tasks, gives different values.

http://docs.ansible.com/ansible/latest/playbooks_lookups.html#the-password-lookup says "A special case is using /dev/null as a path. The password lookup will generate a new random password each time", thus the current behaviour is at least confusing and unexpected.

from irc :

```
<larsks> Richlv: Whatever is happening seems to be happening at a higher level than the password lookup plugin: as far as I can tell, it's only ever getting called once.
```

👍 1

ansibot commented on Dec 21, 2017 Contributor

Files identified in the description:

- test/integration/targets/lookups/aliases

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

ansibot added affects_2.3 bug_report needs_triage support:community labels on Dec 21, 2017

larsks commented on Dec 21, 2017 Contributor

The issue here seems to be template caching. If I modify `lib/ansible/template/__init__.py` so that the `Templar.template` method has `cache=False` instead of `cache=True` as the default, then @richlv's original playbook works as expected.

That means you can work around the problem like this:

```
- hosts: localhost
  gather_facts: false
  tasks:
    - set_fact:
        password1: "{{ lookup('password', '/dev/null length=20') }}"
        password2: "{{ lookup('password', '/dev/null length=20') + '' }}"

    - debug:
        var: password1
    - debug:
        var: password2
```

That gives you two templates that are *functionally* identical but have different content, so caching doesn't bite you.

mkrizek removed the needs_triage label on Dec 22, 2017

ansibot added **bug** and removed **bug_report** labels on Mar 1, 2018

gevial commented on Mar 20, 2018 Contributor

This workaround is troublesome if you need to generate more than 2 passwords though. For example, if you're generating WordPress hash values (https://codex.wordpress.org/Editing_wp-config.php#Security_Keys), you need to generate 8 different values: changing a template 8 times is something you'd want to avoid.

Bugfix would be really helpful.

👍 7

gevial commented on Mar 20, 2018 • edited ▾ Contributor

Anyway, thanks for the finding **@larsks**!

ansibot added the test label on May 23, 2018

ericoc referenced this issue in zeromonio/zeromon on Mar 6, 2019

`specify lengths for generated passwords` 56f8662

ericoc commented on Mar 6, 2019 • edited ▾ Contributor

I had two password lookups in a row that I was leaving with a default `length` of 20 and they were returning identical passwords. Specifying different lengths for each of them cleared it up though and I am now getting unique values: ericoc/zeromon@`56f8662`

```
ansible 2.5.1
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.15rc1 (default, Nov 12 2018, 14:31:15) [GCC 7.3.0]
```

tristanbes commented on Apr 5, 2019 Contributor

This issue is still happening for `ansible 2.7.8`

```
- name: Generate random password for WordPress salts
  set_fact:
    wp_auth_key: "{{ lookup('password', '/dev/null length=64 chars=ascii_letters,digits,punctuation') }}"
    wp_secure_auth_key: "{{ lookup('password', '/dev/null length=64 chars=ascii_letters,digits,punctuation') }}"
    wp_logged_in_key: "{{ lookup('password', '/dev/null length=64 chars=ascii_letters,digits,punctuation') }}"
    wp_nonce_key: "{{ lookup('password', '/dev/null length=64 chars=ascii_letters,digits,punctuation') }}"
    wp_auth_salt: "{{ lookup('password', '/dev/null length=64 chars=ascii_letters,digits,punctuation') }}"
    wp_secure_auth_salt: "{{ lookup('password', '/dev/null length=64 chars=ascii_letters,digits,punctuation') }}"
    wp_logged_in_salt: "{{ lookup('password', '/dev/null length=64 chars=ascii_letters,digits,punctuation') }}"
    wp_nonce_salt: "{{ lookup('password', '/dev/null length=64 chars=ascii_letters,digits,punctuation') }}"
  when: scalingo_existing_wp_salt | length == 0
```

outputs

```
 // outputs
  ok: [127.0.0.1] => {
    "ansible_facts": {
        "wp_auth_key": "%2,+)*%{+`~lV.;u6\\-=-2>X17<|w!5sJiOXJAwH)@<MteGQ<zZ~QRf4O$Z1&[/<",
        "wp_auth_salt": "%2,+)*%{+`~lV.;u6\\-=-2>X17<|w!5sJiOXJAwH)@<MteGQ<zZ~QRf4O$Z1&[/<",
        "wp_logged_in_key": "%2,+)*%{+`~lV.;u6\\-=-2>X17<|w!5sJiOXJAwH)@<MteGQ<zZ~QRf4O$Z1&[/<",
        "wp_logged_in_salt": "%2,+)*%{+`~lV.;u6\\-=-2>X17<|w!5sJiOXJAwH)@<MteGQ<zZ~QRf4O$Z1&[/<",
        "wp_nonce_key": "%2,+)*%{+`~lV.;u6\\-=-2>X17<|w!5sJiOXJAwH)@<MteGQ<zZ~QRf4O$Z1&[/<",
        "wp_nonce_salt": "%2,+)*%{+`~lV.;u6\\-=-2>X17<|w!5sJiOXJAwH)@<MteGQ<zZ~QRf4O$Z1&[/<",
        "wp_secure_auth_key": "%2,+)*%{+`~lV.;u6\\-=-2>X17<|w!5sJiOXJAwH)@<MteGQ<zZ~QRf4O$Z1&[/<",
        "wp_secure_auth_salt": "%2,+)*%{+`~lV.;u6\\-=-2>X17<|w!5sJiOXJAwH)@<MteGQ<zZ~QRf4O$Z1&[/<"
    },
    "changed": false
}
```

I tried also:

```
- name: Generate random password for WordPress salts
  set_fact:
    wp_auth_key: "{{ random_password }}"
    wp_secure_auth_key: "{{ random_password }}"
    wp_logged_in_key: "{{ random_password }}"
    wp_nonce_key: "{{ random_password }}"
    wp_auth_salt: "{{ random_password }}"
    wp_secure_auth_salt: "{{ random_password }}"
    wp_logged_in_salt: "{{ random_password }}"
    wp_nonce_salt: "{{ random_password }}"
  vars:
    random_password: "{{ lookup('password', '/dev/null length=64 chars=ascii_letters,digits,punctuation') }}"
  when: scalingo_existing_wp_salt | length > 0
```

which gives the same password too.

iTaybb commented on Jul 3, 2019

Happens here as well (version 2.8.0).

nicolas-marcq commented on Jul 5, 2019 • edited ▾

And still with version 2.8.2.
Any workaround?

johnhckuo commented on Jul 24, 2019

issue still persist in ansible 2.8.2.post0

hugonz commented on Aug 6, 2019

Also, generation of the same password happens if you specify a different character category, giving a password different than specified.

```
tasks:
   - debug:
       msg: "{{ lookup('password', 'dev/null length=3 chars=ascii_uppercase') }}"
   - debug:
       msg: "{{ lookup('password', 'dev/null length=3 chars=ascii_lowercase') }}"
```

Gives:

```
TASK [debug] ************************************************************************************************************************************
task path: /home/cvx_admin_user/Tests/37/main.yml:7
ok: [localhost] => {
    "msg": "KKU"
}

TASK [debug] ************************************************************************************************************************************
task path: /home/cvx_admin_user/Tests/37/main.yml:9
ok: [localhost] => {
    "msg": "KKU"
}
```

alanbchristie commented on Nov 19, 2019

Just fell into this trap with Ansible `2.9.0` . Any advice on a fix or work-around when generating a number of passwords?

alanbchristie pushed a commit to InformaticsMatters/squonk that referenced this issue on Nov 19, 2019

`- Workaround for Ansible's template cache bug` … `fea616e`

nicolas-marcq commented on Nov 20, 2019

@alanbchristie you can add a blank string with `+ ''` . It work if you call the line multiple time independently. I haven't tried in a loop.

```
"{{ lookup('password', '/dev/null length=15 chars=ascii_letters,digits,._-!')  + '' }}"
```

alanbchristie commented on Nov 20, 2019

Thanks and it's good you've confirmed it works even with more than two instances. I create about 12 passwords in the same file (not in a loop).

To be honest it's a mess which also requires you to leave comments in the file to explain to others who read it why you're doing such an odd thing.

Now I know there's template caching going on, surely, the ultimate solution lies in the engine?

1. Code in the templater to detect these 'corner cases' and *always* re-evaluate, or...
2. A command-line option in ansible to disable templating
3. An annotation in the file to disable templating
4. If this isn't acceptable then switch templating off by default and force users to switch it on in the `ansible.cfg`

But a neat vulnerability that can be exploited! ... If you know something's been deployed using Ansible and you know one password then your best approach for an *attack* is to try the password you know in other apps that have been deployed at the same time. I was deploying 3 independent applications and all 12 passwords ended up being the same! Luckily I spotted it before I rolled it out.

patrick-fls commented on Dec 30, 2019

Same problem... and adding `+ ''` doesn't work in my case:

```
---
- name: "Debug passwords"
  debug:
    msg:
      - "{{lookup('password','/dev/null chars=ascii_letters,digits length=32') + '' }}"
      - "{{lookup('password','/dev/null chars=ascii_letters,digits length=32') + '' }}"
```

yeilds:

```
TASK [test : Debug passwords] ************************************************************************************************************************************************************
ok: [testansible] => {
    "msg": [
        "0te74izZXnhZoU1smcB1b1dYc01uT7Tm",
        "0te74izZXnhZoU1smcB1b1dYc01uT7Tm"
    ]
}
```

alanbchristie commented on Dec 30, 2019

Hi **@patrick-fls**, if you do `+ ''` for just one of them I think it should work. e.g.: -

```
---
- name: "Debug passwords"
  debug:
    msg:
      - "{{lookup('password','/dev/null chars=ascii_letters,digits length=32') }}"
      - "{{lookup('password','/dev/null chars=ascii_letters,digits length=32') + '' }}"
```

It's about *fooling* the template cache algorithm. If the two lines look the same a re-evaluation won't be done.

But (for me) this is just a work-around that woks in only a simple case - i.e. with two passwords because, for the same `chars` and `length` combination it can only be done once. I eventually had to use different values of `length` when I was generating 12 passwords as the `+ ''` hack is of no use for multiple passwords.

I really think the caching algorithm is defeating its purpose here and manifests itself as a **bug**. Personally I think it needs to be fixed or template caching disabled by default as it's too difficult to realise that you've been caught by this **bug** without inspecting the generated variables.

patrick-fls commented on Dec 30, 2019

You're right, it only works for **TWO** passwords. I naively thoughts that doing a concatenation would defeat the cache. I even tried to use `pipe` instead of `password` but same problem. The caching is so overly aggressive....

I'm in a situation where I need several randomly generated 64 byte keys, i can't play with the length. I will have to do it manually on the host instead. Boggles my mind that this is even problem on such a huge system.

alanbchristie commented on Dec 30, 2019

> Boggles my mind that this is even problem on such a huge system.

Agreed - it's one heck of a flaw! ... and no-one is assigned to this so don't expect a resolution anytime soon!

taisph commented on Jan 24, 2020

This is what I ended up with.

```
---
- set_fact:
    deployment_config_template: "{{ deployment_config_template | combine({item: lookup('password', '/dev/null length=42')}) }}"
  loop:
  - sql_password
  - secret_key
  - otherdb_password
  - door_password
  - secret_passphrase
  - ymmv_passphrase
```

felixfontein mentioned this issue on Feb 14, 2020

Templating: make sure only one variable results are cached #67429

Merged

ansibot commented on Feb 17, 2020 Contributor

Files identified in the description:
None

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

ansibot added **support:core** has_pr and removed **support:community** labels on Feb 17, 2020

s-hertel closed this as completed in #67429 on Feb 19, 2020

ansible locked and limited conversation to collaborators on Mar 18, 2020

Assignees

No one assigned

Labels

affects_2.3 bug has_pr support:core test

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Templating: make sure only one variable results are cached
felixfontein/ansible

14 participants