hashicorp / **vault**

"child policies must be subset of parent" #1597

New issue

Closed **eherot** opened this issue on Jul 7, 2016 · 12 comments

eherot commented on Jul 7, 2016 Contributor

Is this really the most logical policy decision?

Here's what I'd like to do (which this check is preventing): I have a worker node whose job is to automatically renew some tokens for my app servers. The accessor token used by this worker has two policies attached to it: worker and default. The worker policy has privileges for "token/create." The token it needs to create has the app and default policies attached.

So even though from a pure security standpoint this kind of makes sense, because obviously if you can create a token with any policy, you can just grant yourself permission to anything else on the system, but as a practical matter it's kind of confusing if the accessor token for my worker box also has to have a bunch of (seemingly unrelated) policies attached to it.

Perhaps I'm confused. Is this exactly the situation that the "sudo" policy was designed to work around?

jefferai commented on Jul 7, 2016 Owner

@eherot `sudo` is a non-granular large hammer for a small nail. But it seems like your exact use-case is already covered by token auth backend roles (https://www.vaultproject.io/docs/auth/token.html).

jefferai closed this on Jul 7, 2016

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

2 participants

eherot commented on Jul 7, 2016 Contributor

Interesting. It kind of seems like roles and policies are two concepts that could actually be merged...

jefferai commented on Jul 7, 2016 Owner

Roles are common concepts in many Vault auth and secret backends. Policies are specific to tokens.

eherot commented on Jul 7, 2016 Contributor

Seems like it would be easier to just have a subkey on a role called "acl" or "permissions" or something like that and have it only be used when the role is assigned to a token (though I'd imagine since it is just a general way of granting access to parts of the system it would have other potential users down the road).

eherot commented on Jul 7, 2016 Contributor

It also did not work. I created a new role called "worker" with the ability to create tokens with many different policies:

```
POST /v1/auth/token/roles/worker HTTP/1.1
X-Vault-Token: ***** Hidden credentials *****
Content-Type: application/json; charset=utf-8
Host: vault.service.prod-us-east-1.consul:8200
Connection: close
User-Agent: Paw/2.3.4 (Macintosh; OS X/10.11.4) GCDHTTPRequest
Content-Length: 71

{"allowed_policies":"apps,default,developers,devops,sf-updater,worker"}
```

Response:

```
HTTP/1.1 204 No Content
Content-Type: application/json
Date: Thu, 07 Jul 2016 18:12:47 GMT
Connection: close
```

Then I created a new token under this role:

```
POST /v1/auth/token/create/worker HTTP/1.1
Content-Type: application/json
X-Vault-Token: ***** Hidden credentials *****
Host: vault.service.prod-us-east-1.consul:8200
Connection: close
User-Agent: Paw/2.3.4 (Macintosh; OS X/10.11.4) GCDHTTPRequest
Content-Length: 103

{
  "policies": [
    "worker"
  ],
  "no_parent": true,
  "ttl": "8760h",
  "display_name": "worker"
}
```

Response (204 - Here's the JSON portion):

```
{
  "lease_id": "",
  "renewable": false,
  "lease_duration": 0,
  "data": null,
  "warnings": null,
  "auth": {
    "client_token": "85915cfc-XXX",
    "accessor": "XXX",
    "policies": [
      "default",
      "worker"
    ],
    "metadata": null,
    "lease_duration": 2592000,
    "renewable": true
  }
}
```

But when I then try to use this token to generate new tokens with different policies:

```
POST /v1/auth/token/create HTTP/1.1
Content-Type: application/json
X-Vault-Token: 85915cfc-XXX
Host: vault.service.prod-us-east-1.consul:8200
Connection: close
User-Agent: Paw/2.3.4 (Macintosh; OS X/10.11.4) GCDHTTPRequest
Content-Length: 107

{
  "policies": [
    "apps"
  ],
  "no_parent": true,
  "ttl": "876h",
  "display_name": "test-20160707"
}
```

Response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Date: Thu, 07 Jul 2016 18:24:03 GMT
Content-Length: 55
Connection: close

{"errors":["child policies must be subset of parent"]}
```

Am I missing something?

jefferai commented on Jul 7, 2016 • edited

Owner

@eherot Token roles are a programmatic way of allowing a token with access to call the create endpoint with that role to request tokens with a subset of the policies in the role's `allowed_policies`. The tokens created against that policy are then subject to normal rules, including children only having a subset of their parents' policies. Example:

```
$ vault write auth/token/roles/worker allowed_policies="apps,default,developers,devops,sf-
updater,worker"
Success! Data written to: auth/token/roles/worker

# No policies selected, gives all policies in Vault 0.6
$ vault token-create -role=worker
Key             Value
---             -----
token           2780edba-c89e-0d23-17f0-da5406dfd902
token_accessor  c69faa1d-b94d-970b-0bd8-d341a5760b72
token_duration  2592000
token_renewable true
token_policies  [apps default developers devops sf-updater worker]

# Selecting only a subset
$ vault token-create -role=worker -policy=apps -policy=devops
Key             Value
---             -----
token           cb4a3c26-282f-f9f3-c5e8-dcd1cfe603da
token_accessor  b015ddad-e1c7-c07a-b8d4-62218b749172
token_duration  2592000
token_renewable true
token_policies  [apps default devops]

# Cannot ask for policies not a subset of the allowed
$ vault token-create -role=worker -policy=foo -policy=bar
```

```
Error creating token: Error making API request.

URL: POST http://127.0.0.1:8200/v1/auth/token/create/worker
Code: 400. Errors:

* token policies ([bar default foo]) must be subset of the role's allowed policies ([apps default
developers devops sf-updater worker])
```

eherot commented on Jul 7, 2016 Contributor

So it sounds like the problem with my above attempt is the the token I tried to create was not also part of the worker role, correct?

jefferai commented on Jul 7, 2016 Owner

No. Tokens are not part of roles. Roles are a feature of the token auth backend that allow you to issue tokens with certain options that are otherwise not normally available to non- `sudo` / `root` users -- like escaping the strict subset checking.

You can use the role to create tokens with different sets of policies than your token, but the issued tokens are then normal tokens with the same matching criteria on their children.

eherot commented on Jul 7, 2016 Contributor

Right, I think I phrased that confusingly. Because I didn't add `/worker` to the request URL for my second token (with the "apps") policy, the request failed, because the role isn't a property of the accessor token, but instead it's a superset where everything contained within it has a certain set of rules applied to it.

Basically it still sounds like the only way to do what I need to do above (where *one* specific token needs to be able to create tokens with different policies) will require sudo.

jefferai commented on Jul 7, 2016 Owner

> Basically it still sounds like the only way to do what I need to do above (where one specific token needs to be able to create tokens with different policies) will require sudo.

You can give the one specific token access to either a single role with the set of desired `allowed_policies`, or multiple roles with the desired sets, and have that single token just create other tokens against those roles.

eherot commented on Jul 7, 2016 Contributor

Thanks for your help, and for letting me draw this out in a closed ticket. I've often found reading the closed tickets here to be helpful at answering my questions so I wanted to do my part to get my question answered in detail. ;-)

👍 1

jefferai commented on Jul 7, 2016 Owner

No problem. GitHub tickets tend not to be great ways to answer questions because of the fact that tickets only have two states and there are many less people with experience that look at them, so in the future if you put questions on here I may direct you back to the mailing list. But I wanted to get you some good state. :-)

♡1