Commit 8b83efa

jechol authored on Oct 17, 2025 · ✕ 47 / 62 · Verified

```
Merge commit from fork

The Policy Refactoring (#2365) introduced a bug where bypass policies
were contributing only their condition to one_condition_matches instead
of their complete expression (condition AND policies).

This caused bypass policies to incorrectly satisfy the "at least one
policy applies" requirement when their condition evaluates to true
but their authorization checks fail.

For example, with these policies:

  bypass always(), do: authorize_if(actor_attribute_equals(:is_admin, true))
  policy action_type(:read), do: authorize_if(always())

The final authorization decision is: one_condition_matches AND all_policies_match

When a bypass condition is true but bypass policies fail, and subsequent
policies have non-matching conditions:

1. one_condition_matches = cond_expr (bypass condition) = true
   (bug - should check if bypass actually authorizes)
2. all_policies_match = (complete_expr OR NOT cond_expr) for each policy
   - For non-matching policies: (false OR NOT false) = true (policies don't apply)
3. Final: true AND true = true (incorrectly authorized)

The bypass condition alone satisfies "at least one policy applies" even
though the bypass fails to authorize.

The fix ensures bypass policies contribute their complete expression
(condition AND policies) to one_condition_matches.
```

main · v3.24.2 ••• v3.7.1

1 parent b2e4d62 commit 8b83efa

2 files changed +65 -8 lines changed

Top

Filter files…

- lib/ash/policy
  - policy.ex
- test/policy
  - policy_test.exs

2 files changed +65 -8 lines changed

Search within code

lib/ash/policy/policy.ex

```
@@ -49,26 +49,25 @@ defmodule Ash.Policy.Policy do
49 49          {policy, cond_expr, complete_expr}
50 50        end)
51 51        |> List.foldr({false, true}, fn
52    -        {%{bypass?: true}, cond_expr, complete_expr}, {one_condition_matches, true} ->
   52 +        {%{bypass?: true}, _cond_expr, complete_expr}, {one_condition_matches, true} ->
53 53            {
54    -            b(cond_expr or one_condition_matches),
55    -            # Bypass can't relay to the next bypass if there is none
   54 +            b(complete_expr or one_condition_matches),
56 55            complete_expr
57 56          }
58 57
59 58        {%FieldPolicy{bypass?: true}, true, complete_expr},
60 59        {one_condition_matches, all_policies_match} ->
61 60          {
62    -            # FieldPolicy Conditions are set to true by default and therefore
63    -            # have to be ignore to not change the meaning of the
64    -            # one_condition_matches condition
65 61            one_condition_matches,
66 62            b(complete_expr or all_policies_match)
67 63          }
68 64
69    -        {%{bypass?: true}, cond_expr, complete_expr}, {one_condition_matches, all_policies_match} ->
   65 +        {%{bypass?: true}, _cond_expr, complete_expr},
   66 +        {one_condition_matches, all_policies_match} ->
70 67          {
```

```
71      -           b(cond_expr or one_condition_matches),
        68  +       # Bypass should only contribute to "at least one policy applies" if it actually authorizes.
        69  +       # Use complete_expr (condition AND policies) not just condition.
        70  +       b(complete_expr or one_condition_matches),
72      71          b(complete_expr or all_policies_match)
73      72        }
74      73
```

test/policy/policy_test.exs

```
        @@ -912,4 +912,62 @@ defmodule Ash.Test.Policy.Policy do
912     912             """
913     913       end
914     914     end
        915 +
        916 +   describe "bypass policy with always-true condition" do
        917 +     test "denies when bypass condition is true but bypass policies fail" do
        918 +       # Scenario: non-admin user performs create action with these policies:
        919 +       #
        920 +       #   bypass always() do
        921 +       #     authorize_if actor_attribute_equals(:is_admin, true)
        922 +       #   end
        923 +       #
        924 +       #   policy action_type(:read) do
        925 +       #     authorize_if always()
        926 +       #   end
        927 +       #
        928 +       # Before fix: Used bypass condition (always) for one_condition_matches
        929 +       #   → one_condition_matches = true (bypass condition matched)
        930 +       #   → all_policies_match = true (no applicable policy)
        931 +       #   → Final: true AND true = true (WRONG! - incorrectly authorized)
        932 +       #
        933 +       # After fix: Use bypass complete_expr (always AND is_admin) for one_condition_matches
        934 +       #   → one_condition_matches = false (bypass failed to authorize)
        935 +       #   → all_policies_match = true (no applicable policy)
        936 +       #   → Final: false AND true = false (CORRECT! - properly denied)
        937 +
        938 +       policies = [
```

```
939 +         # bypass always(), do: authorize_if(actor_attribute_equals(:is_admin,
      true))
940 +         %Ash.Policy.Policy{
941 +           bypass?: true,
942 +           condition: [{Ash.Policy.Check.Static, result: true}],
943 +           policies: [
944 +             %Ash.Policy.Check{
945 +               type: :authorize_if,
946 +               check: {Ash.Policy.Check.Static, result: false},
947 +               check_module: Ash.Policy.Check.Static,
948 +               check_opts: [result: false]
949 +             }
950 +           ]
951 +         },
952 +         # policy action_type(:read) do: authorize_if(always())
953 +         %Ash.Policy.Policy{
954 +           bypass?: false,
955 +           condition: [{Ash.Policy.Check.Static, result: false}],
956 +           policies: [
957 +             %Ash.Policy.Check{
958 +               type: :authorize_if,
959 +               check: {Ash.Policy.Check.Static, result: true},
960 +               check_module: Ash.Policy.Check.Static,
961 +               check_opts: [result: true]
962 +             }
963 +           ]
964 +         }
965 +       ]
966 +
967 +       expression = Ash.Policy.Policy.expression(policies, %{})
968 +
969 +       # Should deny: bypass failed and no policy condition matched
970 +       assert expression == false
971 +     end
972 +   end
915 973     end
```

Comments 0

Please sign in to comment.