Authorization bypass when bypass policy condition evaluates to true

High **zachdaniel** published **GHSA-pcxq-fjp3-r752** on Oct 17, 2025

Package

ash (Hex)

Affected versions

>= 3.6.3 and <= 3.7.0

Patched versions

3.7.1

Description

Summary

Bypass policies incorrectly authorize requests when their condition evaluates to true but their authorization checks fail and no other policies apply.

Impact

Resources with bypass policies can be accessed without proper authorization when:

- Bypass condition evaluates to true
- Bypass authorization checks fail
- Other policies exist but their conditions don't match

Details

Vulnerable code in: lib/ash/policy/policy.ex:69

```
{%{bypass?: true}, cond_expr, complete_expr}, {one_condition_matches, all_policies_h
  {
    b(cond_expr or one_condition_matches),  # <- Bug: uses condition only
    b(complete_expr or all_policies_match)
  }
```

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.

Fix

Replace `cond_expr` with `complete_expr` on line 69:

```
{%{bypass?: true}, _cond_expr, complete_expr}, {one_condition_matches, all_policie c
  {
    b(complete_expr or one_condition_matches),  # <- Fixed
    b(complete_expr or all_policies_match)
  }
```

Line 52 should also be updated for consistency (though it's only triggered when bypass is the last policy, making it coincidentally safe in practice):

```
{%{bypass?: true}, _cond_expr, complete_expr}, {one_condition_matches, true} ->
  {
    b(complete_expr or one_condition_matches),  # <- For consistency
    complete_expr
  }
```

PoC

```
policies do
  bypass always() do
    authorize_if actor_attribute_equals(:is_admin, true)
  end

  policy action_type(:read) do
    authorize_if always()
  end
end
```

Non-admin user can perform create actions (should be denied).

Test demonstrating the bug:

```elixir
test "bypass policy bug" do
  policies = [
    %Ash.Policy.Policy{
      bypass?: true,
      condition: [{Ash.Policy.Check.Static, result: true}],  # condition = true
      policies: [
        %Ash.Policy.Check{
          type: :authorize_if,
          check: {Ash.Policy.Check.Static, result: false},  # policies = false
          check_module: Ash.Policy.Check.Static,
          check_opts: [result: false]
        }
      ]
    },
    %Ash.Policy.Policy{
      bypass?: false,
      condition: [{Ash.Policy.Check.Static, result: false}],
      policies: [
        %Ash.Policy.Check{
          type: :authorize_if,
          check: {Ash.Policy.Check.Static, result: true},
          check_module: Ash.Policy.Check.Static,
          check_opts: [result: true]
        }
      ]
    }
  ]

  expression = Ash.Policy.Policy.expression(policies, %{})

  assert expression == false
  # Expected: false (deny)
  # Actual on main: true (incorrectly authorized)
end
```

Severity

High 8.1 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Unchanged
Confidentiality	High

Integrity	High
Availability	None

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N

CVE ID

CVE-2025-48044

Weaknesses

▶ CWE-285

Credits

jechol	Reporter
maennchen	Analyst
zachdaniel	Remediation reviewer