ssrf

Open

Analyzed | ⋮ | Ignore... | Fix

sims-web_app/Services/BackendApiHandler.cs:93

Assistant thinks this is likely a false positive

This is a false positive. The RestClient is constructed using a constant baseUrl from an environment variable concatenated with a fixed path string. The 'id' parameter is only used within the URL path portion (not controlling the host/domain), which is a normal API usage pattern and not an SSRF vulnerability.

Agree and Ignore or Disagree

RULE DETAILS

- High severity
- Low confidence
- Monitor
- CWE-918
- ssrf

FINDING DETAILS

- 3 days ago
- is241307/simsfh_ws25_SAST
  managed-scan
- main
- 5cef085
  by puywei

Finding description | Rule description

GetIncidentById(string id) builds a RestClient with a URL that directly embeds id: new RestClient(baseUrl + $"/Incidents/{id}"). This lets an attacker control the server-side request path/query your app makes.
Plausible exploit (path/query injection SSRF):

- Assume baseUrl = https://backend.internal
- Attacker supplies id = "%2e%2e%2fadmin%2fsecrets?dump=true"
- Your code constructs and calls: https://backend.internal/Incidents/%2e%2e%2fadmin%2fsecrets?dump=true
- Many servers normalize this to https://backend.internal/admin/secrets?dump=true, so your server makes a request to an internal admin endpoint. Even if response.Data fails to deserialize to Incident, the request still runs, enabling blind SSRF (e.g., probing internal routes or triggering sensitive actions) via repeated id values.

Code involved: GetIncidentById, id, baseUrl, RestClient, RestRequest, ExecuteAsync<Incident>.

Your code | Example code

sims-web_app/Services/BackendApiHandler.cs:93

```
88              return response.Data;
89          }
90          return null;
91      }
92
93      public async Task<Incident?> GetIncidentById(string id)
94      {
95          if (await GetIsUserValid())
96          {
97              RestClient client = new RestClient(baseUrl + $"/Incidents/{id}");
98              RestRequest request = new RestRequest("");
99              RestResponse<Incident> response = await client.ExecuteAsync<Incident>(request);
100
101             return response.Data;
102         }
103         return null;
104     }
105
106     public async Task<bool> DeleteIncident(string id)
107     {
108         if (await GetIsUserValid())
```

Assistant suggested fix | Rule fix

Semgrep Assistant analyzed this finding, but doesn't recommend a fix because it's likely a false positive.

Activity

New note

Assistant did not find any risk markers in this file
3 days ago by Semgrep Assistant

Assistant thinks this is a false positive
This is a false positive. The RestClient is constructed using a constant baseUrl from an environment variable concatenated with a fixed path string. The 'id' parameter is only used within the URL path portion (not controlling the host/domain), which is a normal API usage pattern and not an SSRF vulnerability.
Agree and Ignore | Disagree
3 days ago by Semgrep Assistant

Assistant analysis started
Generating triage recommendations, component tags, and remediation advice
3 days ago by Gabschgo

Seen on
main
Wed, 19 Nov 2025 15:22:11
3 days ago via GitHub