Closed · Issue created 2 years ago by GitLab SecurityBot

SSRF: send arbitrary POST requests into local networks

HackerOne report #1037411 by yvvdwf on 2020-11-18, assigned to @dcouture:

Report | How To Reproduce

Report

Hello,

Prometheus integration service allows to issue any POST requests into local network.

Steps to reproduce

1. In any existing project (or create a new one), goto Settings/Intergrations/Prometheus
2. Setting the following parameters:

- Active: check
- API URL: `https://gitlab.com` (or any url)
- Google IAP Audience Client ID: `test` (or any string)
- Google IAP Service Account JSON: `{"token_credential_uri": "http://localhost:1234/arbitrary-query?x=y", "private_key": 512}`

3. Click "Test" button, gitlab will hit `localhost:1234` (e.g., open a web server at port 1234 to receive the requests)

Impact

The SSRF allows attackers to issue arbitrary POST requests into local network. The impact will be different, depending on which services are running on the local network. For example, if there exists a docker daemon that is listening on localhost:2375, then attackers may read any files by issuing a POST request to build an image from remote dockerfile. With your permission, I will try to discover gitlab.com .

Output of checks

This bug happens on GitLab.com

Impact

The SSRF allows attackers to issue arbitrary POST requests into local network. The impact will be different, depending on which services are running on the local network. For example, if there exists a docker daemon that is listening on localhost:2375, then attackers may read any files by issuing a POST request to build an image from remote dockerfile. With your permission, I will try to discover gitlab.com .

How To Reproduce

Please add reproducibility information to this section:

1.
2.
3.

Edited 1 year ago by Dominic Couture

Drag your designs here or click to upload.

Tasks 0

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 0

Link issues together to show that they're related or that one is blocking others. Learn more.

Related merge requests 1

Support IAP protected prometheus installations
!33508 · 13.1

Activity

GitLab SecurityBot changed due date to January 17, 2021 2 years ago

GitLab SecurityBot added Weakness::CWE-918 priority::2 severity::2 scoped labels 2 years ago

GitLab SecurityBot added HackerOne security labels 2 years ago

Dominic Couture @dcouture · 2 years ago (Developer)

Hello @mnohr 👋! We received a report for a POST SSRF vulnerability in the prometheus integration.

This appears to be because of the `googleauth` gem that obviously doesn't use `GitLab::HTTP` .

https://gitlab.com/gitlab-org/gitlab/-/blob/c39c15466fe929a22b070d6a3adc3024c5551fde/app/models/project_services/prometheus_service.rb#L187

Dominic Couture @dcouture · 2 years ago (Developer)

The reporter added that `Google::Auth::Credentials.new` also accepts a filename as an argument and while it doesn't seem to be exploitable at the moment, we should probably block that and make sure the parsed JSON `.is_a?(Hash)` .

> Hi Dominic,
>
> Thank you for triaging. Otherwise, going further in google auth method, I found a *potential* vulnerable that can occur on gitlab and I think that gitlab should fix it.
>
> Prometheus service calls `Google::Auth::Credentials.new` by giving a hash object that is generated from JSON string given by users via `google_iap_service_account_json` variable.
>
> ```
> def iap_client
>   @iap_client ||= Google::Auth::Credentials.new(Gitlab::Json.parse(google_iap_service_account_json), t
> end
> end
> ```
>
> In fact, the function `Google::Auth::Credentials.new` accepts also a string as file name, then it will read json content from that file. Imagine if we set value of **Google IAP Service Account JSON** to `"/tmp/tata"` (the opened and closed quotes are important as `Gitlab::Json.parse` will generate a string)... If the file does not validate json format, an exception will be thrown with content of the file.
>
> Currently the exception is printed out to execution log of gitlab. Thus, it is only a *potential* vuln (and we know what happens if attackers are able to find an endpoint to request prometheus service and it returns the bug message..)
>
> Regrads,

Please register or sign in to reply

Dominic Couture added group::ecosystem [DEPRECATED] devops::create scoped labels 2 years ago

GitLab SecurityBot @gitlab-securitybot · 2 years ago (Author) (Reporter)

@deuley @leipert This issue is ready for triage as per HackerOne process.

About this automation: AppSec Escalation Engine

GitLab SecurityBot added security-set-milestone label 2 years ago

GitLab Bot added section::dev scoped label 2 years ago

GitLab SecurityBot @gitlab-securitybot · 2 years ago (Author) (Reporter)

@deuley This severity 2 security issue has no milestone yet. (For remediation goals, please see Severity and Priority Labels on --security Issues.)

About this automation: AppSec Escalation Engine

Lukas 'Eipi' Eipert @leipert · 1 year ago — Developer

@mnohr If it relates to the Prometheus integrations it sounds like --"group::monitor".

Bringing this to your attention. @crystalpoole!

Lukas 'Eipi' Eipert added group respond devops monitor scoped labels and automatically removed group ecosystem [DEPRECATED] devops create labels 1 year ago

---

Crystal Poole @crystalpoole · 1 year ago — Developer

@ck3g @vdesousa Can you triage this and decide how it can be resolved? @splattael has ideas for a couple of options.

@sarahwaldner I'm adding this to %13.9.

Sarah Waldner @sarahwaldner · 1 year ago — Developer

Thank you @crystalpoole

Peter Leitzen @splattael · 1 year ago — Maintainer

@crystalpoole Thanks for pinging 🙏

My ideas were:

- Validate `token_credential_uri` the same we're doing with `Gitlab::Http`
  - Maybe it's worth extracting this code to encourage use for future cases
- Check if IAP JSON input is a Hash to prevent users passing a string which is treated a filename

---

Dominic Couture @dcouture · 1 year ago — Developer

Because the actual HTTP request doesn't come from our code, the challenge here will be to avoid a Time of Check/Time of Use bug where the attacker could use DNS rebinding to have a valid non-local URL when we validate it but change the DNS record to an internal IP when the library issues the request.

I'm aware I'm just dropping this without offering a solution, I'll keep thinking about it and try to contribute something more helpful if I have a good idea. :)

Peter Leitzen @splattael · 1 year ago — Maintainer

@dcouture Thanks for jumping on a call with me and clarifying 🙏

Validating the `token_credential_uri` before handing it off to the googleauth gem does not prevent DNS rebinding because we'd need to reuse the resolved IP (see https://gitlab.com/gitlab-org/gitlab/-/blob/54b1afcb381d7733c1bcfa3947e41eafa08f6350/lib/gitlab/http_connection_adapter.rb#L24) during the HTTP call and we cannot control the gem code.

One idea could be to just block the property `token_credential_uri` to be set by users and exclude it from the JSON. We'd need to check whether users are actually providing this property for good use beforehand though 🥵

@akohlbecker Do you remember use cases where it'd make sense for users to set `token_credential_uri`? 🤔

Peter Leitzen @splattael · 1 year ago — Maintainer

@dcouture It seems that one can configure a connection by passing additional options to `Google::Auth::Credentials.new`. So, there's hope that we can perform the very same checks as well 🤞

Edited by Peter Leitzen 1 year ago

Adrien Kohlbecker @akohlbecker · 1 year ago — Developer

@splattael @dcouture I wasn't really familiar with how this parameter can be used, however I have done some quick research and for reference:

- This is the MR that introduced the functionality in Gitlab !33508 (merged)
- In `google-auth-library-ruby` this is the MR that added support for reading this property from the JSON. (It doesn't have details on why that would be needed, unfortunately) https://github.com/googleapis/google-auth-library-ruby/pull/251. However, this shows that the default behavior when not provided is to use google's token endpoint.
- In `signet` which is the underlying auth gem, they say this property is used to issue tokens and refresh outdated ones automatically, which would explain the outgoing request https://github.com/googleapis/signet/blob/cd5d2792055e99e188b81a4210e717ddb4071fa3/lib/signet/oauth_2/client.rb#L37-L39

My inclination would be to strip this property or maybe prevent the form from being valid if a user has it set. Given that this feature is intended to be used with Google IAP, we probably don't need to support any other token issuer, so resetting it to `google-auth-library-ruby`'s default seems the right call.

I would also second taking a look in the prod DB and see if any user has it set and what values they used, which could inform the decision.

Let me know your thoughts!

Peter Leitzen @splattael · 1 year ago — Maintainer

@akohlbecker Thank you so much for your research 🙏

This is great to know that we potentially can strip this dangerous property since it's much easier than verifying (to protect from DNS rebinding) the URL in the googleauth gem 🙂

On `#database-lab` I've tried to find out who's actually `google_iap_service_account_json` with `token_credential_uri`:

Overall

```
SELECT count(*) FROM "services" WHERE "services"."type" = 'PrometheusService' and properties like '%google_
```

Result: `Aggregate  (cost=43259.43..43259.44 rows=1 width=8) (actual time=48.489..48.491 rows=1 loops=1)`

`rows=1` ❗

With `token_credential_uri`

```
SELECT count(*) FROM "services" WHERE "services"."type" = 'PrometheusService' and properties like '%google_
```

Result: `Aggregate  (cost=43259.40..43259.41 rows=1 width=8) (actual time=38.887..38.889 rows=1 loops=1)`

`rows=1` ❗ ❗

So we have a single user on GitLab.com who uses `google_iap_service_account_json` but also passed `token_credential_uri`. This *might* be the HackerOne reporter 🤔

I'll ask for its contents in `#production`. ⏳

Edit: See https://gitlab.com/gitlab-com/gl-infra/infrastructure/-/issues/12291

Edited by Peter Leitzen 1 year ago

Adrien Kohlbecker @akohlbecker · 1 year ago — Developer

Hey @splattael I just ran your query 🙂

```
[ gprd ] production> res = ActiveRecord::Base.connection.execute(%(SELECT properties FROM "services" WHERE "
=> #<PG::Result:0x00007febbe10a598 status=PGRES_TUPLES_OK ntuples=2 nfields=1 cmd_tuples=2>
[ gprd ] production> res[0]
=> {"properties"=>"{\"api_url\":\"https://gitlab.com\",\"google_iap_service_account_json\":\"{\\\"token_cred
[ gprd ] production> res[1]
=> {"properties"=>"{\"api_url\":\"https://gitlab.com\",\"google_iap_service_account_json\":\"{\\\"token_cred
```

So these seem to be dummy values.

Edited by Peter Leitzen 1 year ago

Peter Leitzen @splattael · 1 year ago — Maintainer

@akohlbecker Thanks a lot for getting this data 🙇

`REDACTED` seems to be connected but does not respond on port 80 🤔 😅

```
$ curl http://REDACTED/
curl: (7) Failed to connect to REDACTED port 80: Connection refused
```

Let's block `token_credential_uri` then 💪

Edited by Peter Leitzen 1 year ago

Dominic Couture @dcouture · 1 year ago — Developer

`yvvdwf` is the username of the person reporting this through HackerOne, definitely test data.

Vitor Meireles De Sousa @vdesousa · 1 year ago — Developer

Thanks @dcouture for handling that while I was out! 👍

Please register or sign in to reply

Crystal Poole changed milestone to %13.9 1 year ago

Crystal Poole mentioned in issue gitlab-org/monitor/monitor#49 1 year ago

GitLab SecurityBot removed security-set-milestone label 1 year ago

GitLab Bot added Accepting merge requests label 1 year ago

Peter Leitzen mentioned in issue gitlab-com/gl-infra/infrastructure#12291 1 year ago

Crystal Poole added workflow::ready for development scoped label 1 year ago

GitLab SecurityBot @gitlab-securitybot · 1 year ago — Author, Reporter

@crystalpoole @ClemMakesApps This severity::2 security issue has an active milestone, but currently no assignee(s).

About this automation: AppSec Escalation Engine

Peter Leitzen assigned to @splattael 1 year ago

Peter Leitzen @splattael · 1 year ago — Maintainer

https://gitlab.com/gitlab-org/security/gitlab/-/merge_requests/1204 attempts to fix this issue.

/cc @dcouture

GitLab Bot removed Accepting merge requests label 1 year ago

Peter Leitzen @splattael · 1 year ago — Maintainer

Async issue update

Please provide a quick summary of the current status (one sentence).

Fix created and MR (target `master`) is in workflow::in review.

When do you predict this feature to be ready for maintainer review?

It is in review and 3 more backport MR to follow after successful approval.

Are there any opportunities to further break the issue or merge request into smaller pieces (if applicable)?

No

Peter Leitzen added workflow::in review scoped label and automatically removed workflow::ready for development label 1 year ago

Costel Maxim @cmaxim · 1 year ago — Developer

Patch released with 13.8.4. Closing this issue. Thanks

Costel Maxim closed 1 year ago

GitLab SecurityBot @gitlab-securitybot · 1 year ago — Author, Reporter

This HackerOne security issue was closed 30 days ago and may become public.

Please ensure the following items are true and add a ✅ reaction:

- Issue description and comments do not contain sensitive data belonging to GitLab.
- Issue does not reveal private information of the reporter (i.e. session IDs, passwords).

If the issue needs to stay confidential, please add the keep confidential label.

If you removed confidential data from the issue description before making it public, make sure that the description history entry is deleted.

Peter Leitzen @splattael · 1 year ago — Maintainer

I've redacted potential private information in #284819 (comment 484809681).

Peter Leitzen @splattael · 1 year ago — Maintainer

Making this issue public.

Please register or sign in to reply

Peter Leitzen made the issue visible to everyone 1 year ago

Dominic Couture made the issue confidential 1 year ago

Dominic Couture made the issue visible to everyone 1 year ago

Dominic Couture changed the description 1 year ago

Please register or sign in to reply