Closed (moved)   Opened 2 years ago by   **John Northrup**

Brief SSH Outage

Context

An errant option was included in a refactoring of the HA-Proxy configuration which lead to all SSH traffic being wrapped in SSL.

Timeline

On date: 2017-09-11

- 06:28 UTC - HA Proxy is reloaded with config
- 06:30 UTC - Issues begin to be reported
- 06:40 UTC - Errant configuration is noticed in SSH backend
- 06:52 UTC - Fix for Proxy frontends is started to deploy
- 07:02 UTC - Restart of HA Proxy is required to make change effective, restarts issued
- 07:08 UTC - Functionality restord for SSH access
- 07:12 UTC - 0.1.67 of `gitlab-haproxy` is rolled to correct SSL configuration
- 07:23 UTC - All frontend nodes verified to have correct config

Incident Analysis

- How was the incident detected?
  - It was reported by our monitoring, and by users via Twitter and GitLab Issues.
- Is there anything that could have been done to improve the time to detection?
  - Monitoring caught this exception and raised an alert.
- How was the root cause discovered?
  - Diff of running configuration vs previous running configuration.
- Was this incident triggered by a change?
  - Yes
- Was there an existing issue that would have either prevented this incident or reduced the impact?
  - No, this change passed configuration pre-flight checks

Root Cause Analysis

Follow the the 5 whys in a blameless manner as the core of the post mortem.

- Why did this change cause downtime?
  - Because an errant ha proxy verb made it's way into a configuration
- Why wasn't this error caught?
  - Because the level of checking that was done was to make sure HA Proxy pre-flight compilation passed. This passed because there was nothing ȳ ŵ Ö̧ with what was being asked of the configuration, but because the configuration was incorrect for our architecture.
- Why weren't there more checks for real-world access, such as an end-to-end validation that services depending upon the proxy can connect?
  - This is an area we need to write more tests for.

What went well

- Additional functionality was added to the HA Proxy that was expected and unblocks other projects.

What can be improved

- More testing needs to be added to our HA Proxy testing to validate our deploy assumptions.

Corrective actions

- https://gitlab.com/gitlab-com/infrastructure/issues/2742

Guidelines

- Blameless Postmortems Guideline

- [5 whys](#)

Edited 2 years ago by Daniele Valeriani [GitLab]

Linked issues ❓　　🗂 1

Relates to

⊖　ssh_exchange_identification: Connection closed by remote host

gitlab-org/gitlab-foss#1398

John Northrup @northrup · 2 years ago

Within the HA Proxy configuration we use a 'front-end' / 'back-end' configuration, with separate nodes for SSH traffic. Because we tie the services of git via https and git via SSH together, we perform an https service check on the backend host and drop all services (git via https and git via SSH) if the https service is not responsive.

The previous line was

```
server <%= name %> <%= ip %>:22 check check-ssl port 443 inter 5000 fall 3 veri1
```

The errant line that was added was

```
server <%= name %> <%= ip %>:22 check check-ssl port 443 inter 2s fastinter 1s
```

the `ssl` verb at the end erroneously told HA Proxy to wrap all SSH traffic in SSL to the backend git servers.

Daniele Valeriani [GitLab] @omame-gitlab changed the description 2 years ago

Daniele Valeriani [GitLab] @omame-gitlab mentioned in issue #2741 (moved) 2 years ago

Emil Vikström @emilv · 2 years ago

According to https://status.gitlab.com/ the SSH service was down for almost 4 hours, from 4:02 to 7:46. Was there some other error before 6:28?

Shivang Kar @shivang.kar · 2 years ago

The SSH was working fine few minutes after the issue was reported. But I came across 500 and 502 errors while visiting gitlab.com.

Edited by Shivang Kar 2 years ago

Ghost User @ghost1 · 2 years ago

hello @shivang.kar , unfortunately we have had two unrelated outages today. this one and https://gitlab.com/gitlab-com/infrastructure/issues/2744 . The second one caused the 500/502 errors you encountered.

Stan Hu @stanhu · 2 years ago　　　　　　　　　　　　　　　　　Maintainer

@sitschner Do we need to revisit the way changes to the load balancer are deployed to GitLab.com? This seems like it should have been tested in some form before rolled out to the whole fleet (e.g. via staging).

Edited by Stan Hu 2 years ago

Ghost User @ghost1 · 2 years ago

@stanhu yes. We are working on staging, and getting it closer to production. it would be a good place for end to end tests before deploying to prod.

John Northrup @northrup · **2 years ago**

@stanhu I have an open issue, listed in the "corrective actions", to get better end-to-end testing.

Pablo Carranza [GitLab] @pcarranza-gitlab · **2 years ago**

Closing this issue as we have a corrective action.

Pablo Carranza [GitLab] @pcarranza-gitlab closed **2 years ago**

Andrew Newdigate @andrewn moved to production#239 (closed) 1 year ago

Please register or sign in to reply