New issue

Jump to bottom

CVE-2021-25742: Ingress-nginx custom snippets allows retrieval of ingress-nginx serviceaccount token and secrets across all namespaces #7837

Closed   **cjcullen** opened this issue on Oct 21, 2021 · 60 comments

Assignees

Labels   kind/bug   priority/critical-urgent   triage/accepted

cjcullen commented on Oct 21, 2021 • edited   Member

Issue Details

A security issue was discovered in ingress-nginx where a user that can create or update ingress objects can use the custom snippets feature to obtain all secrets in the cluster.

This issue has been rated **High** (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:L), and assigned **CVE-2021-25742**.

Affected Components and Configurations

This bug affects ingress-nginx.

Multitenant environments where non-admin users have permissions to create Ingress objects are most affected by this issue.

Affected Versions with no mitigation

- v1.0.0
- <= v0.49.0

Versions allowing mitigation

This issue cannot be fixed solely by upgrading ingress-nginx. It can be mitigated in the following versions:

- v1.0.1
- v0.49.1

Mitigation

To mitigate this vulnerability:

1. Upgrade to a version that allows mitigation, (>= v0.49.1 or >= v1.0.1)

2. Set allow-snippet-annotations to false in your ingress-nginx ConfigMap based on how you deploy ingress-nginx:

   Static Deploy Files
   Edit the ConfigMap for ingress-nginx **after** deployment:

   ```
   kubectl edit configmap -n ingress-nginx ingress-nginx-controller
   ```

   Add directive:

   ```
   data:
     allow-snippet-annotations: “false”
   ```

   More information on the ConfigMap here

   Deploying Via Helm
   Set `controller.allowSnippetAnnotations` to `false` in the Values.yaml or add the directive to the helm deploy:

   ```
   helm install [RELEASE_NAME] --set controller.allowSnippetAnnotations=false ingress-nginx/ingress-nginx
   ```

   https://github.com/kubernetes/ingress-nginx/blob/controller-v1.0.1/charts/ingress-nginx/values.yaml#L76

Detection

If you find evidence that this vulnerability has been exploited, please contact security@kubernetes.io
Additional Details
See ingress-nginx Issue #7837 for more details.

Acknowledgements

This vulnerability was reported by Mitch Hulscher.

Thank You,
CJ Cullen on behalf of the Kubernetes Security Response Committee

🎉 14   ❤️ 24   👀 8

cjcullen added the kind/bug label on Oct 21, 2021

k8s-ci-robot added the **needs-triage** label on Oct 21, 2021

k8s-ci-robot commented on Oct 21, 2021 Contributor

@cjcullen: This issue is currently awaiting triage.

If Ingress contributors determines this is a relevant issue, they will accept it by applying the `triage/accepted` label and provide further guidance.

The `triage/accepted` label can be added by org members by writing `/triage accepted` in a comment.

▶ Details

k8s-ci-robot added the **needs-priority** label on Oct 21, 2021

cjcullen changed the title ~~HOLD~~ **CVE-2021-25742: Ingress-nginx custom snippets allows retrieval of ingress-nginx serviceaccount token and secrets across all namespaces** on Oct 21, 2021

rikatz commented on Oct 21, 2021 Contributor

/assign

k8s-ci-robot assigned **rikatz** on Oct 21, 2021

rikatz commented on Oct 21, 2021 Contributor

Fixed in 0.X branch -> #7666
Fixed in v1.X branch -> #7670

👍 1

rikatz added priority/critical-urgent **triage/accepted** and removed **needs-triage** labels on Oct 21, 2021

k8s-ci-robot removed the **needs-priority** label on Oct 21, 2021

paolomainardi commented on Oct 21, 2021 • edited

I don't know if it related but our ingresses stopped to work 2 hours ago (more or less) with:

```
Error: UPGRADE FAILED: cannot patch "develop-website-ingress" with kind Ingress: Internal error occurred: failed calling webhook "validate.nginx.ingress.kubernetes.io": Post
"https://ingress-nginx-controller-admission.ingress-nginx.svc:443/networking/v1beta1/ingresses?timeout=10s": service "ingress-nginx-controller-admission" not found
```

And the service was not in place at all, just the `ingress-nginx-admission` was present, i had to manually delete it, the fun fact is that the ingress-nginx installation had the admission webhook disabled, still trying to face what happened here.

GKE - 1.19.13-gke.1200

iamNoah1 commented on Oct 21, 2021 Contributor

@paolomainardi I don't think so. The CVE says, that non admin users who create or update ingress nginx instances, could possibly all secrets of the cluster. Sound more like a bug TBH. Feel free to raise an issue :)

👍 1

paolomainardi commented on Oct 21, 2021

@iamNoah1 yes i guess so, but you know the timing was so close that i thought it was worth sharing, thanks :)

👍 2

Typositoire commented on Oct 21, 2021

Might be obvious to others but there's an actually FIX coming right ? The final fix isn't removing a feature ?

👍 16

dezmodue commented on Oct 21, 2021

Hi, I would appreciate some clarification.

Is the ability to retrieve "secrets across all namespaces" a consequence of "Ingress-nginx custom snippets allows retrieval of ingress-nginx serviceaccount token"?

In other words, the vulnerability allows a user to access the serviceaccount token and as a result of that the service token can then be used to access any resource that the token is allowed?

JimBugwadia mentioned this issue on Oct 21, 2021

policy to check CM and Ingress for nginx custom snippets kyverno/policies#146

Merged

rjes mentioned this issue on Oct 22, 2021

Is nginxinc ingress controller affected by CVE-2021-25742? nginxinc/kubernetes-ingress#2116

Closed

raesene commented on Oct 22, 2021

@dezmodue I had a look at this issue today, and there's definitely an attack path that gets the ingress-nginx service account token, which has list rights on secrets at a cluster level (so allowing for all secret values to be retrieved). There may be other attack paths as well, but that is one of them.

longwuyuan commented on Oct 22, 2021 Contributor

Can you kindly check if disabling snippets helps.

Thanks,
; Long
…

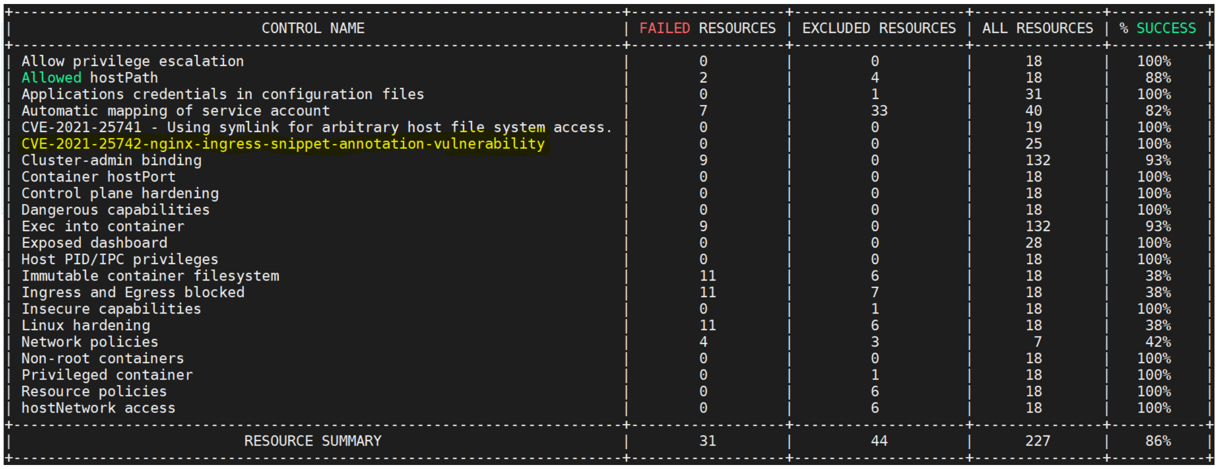
dezmodue commented on Oct 22, 2021

Hi **@raesene**, thanks for your reply.

In some setups the ingress controller is "namespaced" and the service account is only allowed to get secrets that belong to the same namespace. The users with access to the namespace are the ones managing the ingress controller and have access to the token already so this seems to be a non issue in this scenario.
Is my understanding correct?

dwertent commented on Oct 22, 2021

Well Kubescape already came out with a control that will detect whether your cluster has this vulnerability



```
+-----------------------------------------------------------------------+------------------+--------------------+---------------+-----------+
|                             CONTROL NAME                              | FAILED RESOURCES | EXCLUDED RESOURCES | ALL RESOURCES | % SUCCESS |
+-----------------------------------------------------------------------+------------------+--------------------+---------------+-----------+
| Allow privilege escalation                                            |        0         |         0          |      18       |   100%    |
| Allowed hostPath                                                      |        2         |         4          |      18       |    88%    |
| Applications credentials in configuration files                       |        0         |         1          |      31       |   100%    |
| Automatic mapping of service account                                  |        7         |         33         |      40       |    82%    |
| CVE-2021-25741 - Using symlink for arbitrary host file system access. |        0         |         0          |      19       |   100%    |
| CVE-2021-25742-nginx-ingress-snippet-annotation-vulnerability         |        0         |         0          |      25       |   100%    |
| Cluster-admin binding                                                 |        9         |         0          |      132      |    93%    |
| Container hostPort                                                    |        0         |         0          |      18       |   100%    |
| Control plane hardening                                               |        0         |         0          |      18       |   100%    |
| Dangerous capabilities                                                |        0         |         0          |      18       |   100%    |
| Exec into container                                                   |        9         |         0          |      132      |    93%    |
| Exposed dashboard                                                     |        0         |         0          |      28       |   100%    |
| Host PID/IPC privileges                                               |        0         |         0          |      18       |   100%    |
| Immutable container filesystem                                        |        11        |         6          |      18       |    38%    |
| Ingress and Egress blocked                                            |        11        |         7          |      18       |    38%    |
| Insecure capabilities                                                 |        0         |         1          |      18       |   100%    |
| Linux hardening                                                       |        11        |         6          |      18       |    38%    |
| Network policies                                                      |        4         |         3          |       7       |    42%    |
| Non-root containers                                                   |        0         |         0          |      18       |   100%    |
| Privileged container                                                  |        0         |         1          |      18       |   100%    |
| Resource policies                                                     |        0         |         6          |      18       |   100%    |
| hostNetwork access                                                    |        0         |         6          |      18       |   100%    |
+-----------------------------------------------------------------------+------------------+--------------------+---------------+-----------+
|                           RESOURCE SUMMARY                            |        31        |         44         |      227      |    86%    |
+-----------------------------------------------------------------------+------------------+--------------------+---------------+-----------+
```

👍 15 🎉 4

raesene commented on Oct 22, 2021

@dezmodue So for the case I found, I think so. Basically the snippets facility seems to provide effectively full control over the nginx config and access to the ingress-controller container, so if those are effectively not issues (i.e. all the resources owned by the ingress controller are already owned by the individual teams) then it doesn't *seem* like there'd be a big impact (AFAIK and I'm not one of the devs, just a curious security person :))

👍 2

roy-work commented on Oct 22, 2021

What is the actual vuln.?

> See ingress-nginx Issue #7837 for more details.

That link is to this issue, i.e., it links to itself "for more details".

😄 10

rikatz commented on Oct 22, 2021 Contributor

Hi folks, I'm gonna try to answer all your questions here, and we are also scheduling an office hours to explain about this vulnerability and why you should take care of it.:

> **@Typositoire**: Might be obvious to others but there's an actually FIX coming right ? The final fix isn't removing a feature ?
> We didn't discussed actually how to fix this, and it may be hard. The main problem is that the custom snippet annotations allows users to add code that might be executed by Nginx (via Lua) so we should think in a way to map and drop those actions, or create maybe a "safe/non safe" directives that may be dropped when used in annotations.

> **@dezmodue** Is the ability to retrieve "secrets across all namespaces" a consequence of "Ingress-nginx custom snippets allows retrieval of ingress-nginx serviceaccount token"?
> Yes. Ingress nginx demands a full secret access on your cluster, as the TLS certificates that you may point in your ingress object are secrets. So using ingress-nginx service account allows you to query any secret in your cluster (and other stuff, take a look into the RBAC file)

> **@raesene** So for the case I found, I think so. Basically the snippets facility seems to provide effectively full control over the nginx config and access to the ingress-controller container, so if those are effectively not issues (i.e. all the resources owned by the ingress controller are already owned by the individual teams) then it doesn't seem like there'd be a big impact (AFAIK and I'm not one of the devs, just a curious security person :))

Correct. If you trust users with access to the ingress object there is no big deal. The problem here is, if you share a cluster, and some user uses the custom annotation to add random code that can be used to exfiltrate secrets (or do other stuff).

> **@roy-work** What is the actual vuln.?

This is the actual vuln. You can use custom snippets to run arbitrary code and exfiltrate secrets from the container running ingress nginx.

> Got a question in Slack if disabling snippets will disable all of the directives (modsecurity, etc).
> The answer for this is in here but explaining:

- If a USER tries to use some snippet annotation (configuration-snippet, server-snippet, modsecurity-snippet, externalauth-snippet) that annotation gets dropped
- If the cluster admin uses the configmap snippet (like https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/configmap/#main-snippet) then it works. This is because we TRUST on the ingress admin (and the definition to enable annotations is already inside this configmap anyway).

roy-work commented on Oct 22, 2021

> > **@roy-work** What is the actual vuln.?
>
> This is the actual vuln. You can use custom snippets to run arbitrary code and exfiltrate secrets from the container running ingress nginx.

@rikatz I should have been more precise, I guess. I surmised that it was probably the case that some sort of code execution was occurring (others are guessing that that is used to access the service account credentials, and from there, "it's obvious" suffices). My understanding of snippets is that they're nginx configurations; how is it that I can run code from an nginx configuration? (Does nginx have a directive that amounts to an `exec(2)` ?)

Typositoire commented on Oct 22, 2021 • edited ▾

@roy-work

> The main problem is that the custom snippet annotations allows users to add code that might be executed by Nginx (via Lua) so we should think in a way to map and drop those actions, or create maybe a "safe/non safe" directives that may be dropped when used in annotations.

With

> Yes. Ingress nginx demands a full secret access on your cluster, as the TLS certificates that you may point in your ingress object are secrets. So using ingress-nginx service account allows you to query any secret in your cluster (and other stuff, take a look into the RBAC file)

It's not split from the original questions, I guess this was his answer :p

👍 1

rikatz commented on Oct 22, 2021 Contributor

@roy-work I prefer in this case not enter in details, just bare in mind that NGINX does not have this directive, but we use some other modules that may allow the vulnerability written here :)

max0ne mentioned this issue on Oct 22, 2021

Add disallow nginx snippet policy cruise-automation/k-rail#126

Merged

61 hidden items
Load more...

Sh1ftry commented on Nov 22, 2021 • edited ▾

On each sync the blocklist is logged with error level. Shouldn't it have info level and a bit clearer message like `Checking snippet annotations for blocklist: ...` ?

ingress-nginx/internal/ingress/controller/store/store.go
Line 826 in e57d2f6

```
826      klog.Errorf("Blocklist: %v", s.backendConfig.AnnotationValueWordBlocklist)
```

👍 5

dirsigler mentioned this issue on Nov 22, 2021

Admission Webhook denies request dirsigler/uptime-kuma-helm#7

Closed

nnewc mentioned this issue on Nov 22, 2021

Support grpc keep alive server parameters #4402

Closed

mxey mentioned this issue on Nov 29, 2021

Option to disable usage of secrets completely #7990

Closed

raesene commented on Dec 4, 2021

As this is now publicly available, seems like the technical details in the HackerOne report for this issue could be useful for people looking to design/test mitigations
https://hackerone.com/reports/1249583

👍 5

sherifabdlnaby mentioned this issue on Dec 8, 2021

Custom headers via annotation (make it bulletproof instead of using configuration-snippet) #7811

Open

antoineozenne mentioned this issue on Dec 9, 2021

A specific annotation for more_set_input_headers ? #8027

Closed

sdickhoven commented on Dec 13, 2021 • edited

rather than opening a new issue, i'm going to use this one to mention an ambiguity in the documentation.

this pr adds documentation for "suggested" strings that should be blocked in `*-snippet` annotations: #7942

```
  _**suggested:**_ `"load_module,lua_package,_by_lua,location,root,proxy_pass,serviceaccount,{,},',\"`
```

it is not clear whether the last character that should be blocked is a `\` or a `"` since the above string is not really valid in any interpreter. since `'` is excluded, maybe `"` should be excluded too? or is `\` the troublesome character?

maybe the documentation intends to list all the troublesome strings in which case it may be better to do:

```
  _**suggested:**_ "`load_module,lua_package,_by_lua,location,root,proxy_pass,serviceaccount,{,},',\`"

  - OR -

  _**suggested:**_ "`load_module,lua_package,_by_lua,location,root,proxy_pass,serviceaccount,{,},',"`"
```

or

```
  _**suggested:**_ `"load_module,lua_package,_by_lua,location,root,proxy_pass,serviceaccount,{,},',\\"`

  - OR -

  _**suggested:**_ `"load_module,lua_package,_by_lua,location,root,proxy_pass,serviceaccount,{,},',\""`
```

or

```
  _**suggested:**_ `load_module,lua_package,_by_lua,location,root,proxy_pass,serviceaccount,{,},',\`

  - OR -

  _**suggested:**_ `load_module,lua_package,_by_lua,location,root,proxy_pass,serviceaccount,{,},',"`
```

with the double quotes included in the code comment, it seems like this should be a copy-paste-able string for my yaml config. but, of course, the yaml parser will bawk at the unterminated string: `" ... \"` . a valid string would be `" ... \""` or `" ... \\"` .

👍 2

NissesSenap mentioned this issue on Dec 14, 2021

Ingress block annotation XenitAB/terraform-modules#472

Merged

zohebs341 commented on Jan 12 • edited

Hi All,

Do we need to update configmap, even if nginx version is 1.19.4

As I'm expecting, by default below parameter will be false in nginx 1.19.4

allow-snippet-annotations:"false"

appreciate your inputs

morriq mentioned this issue on Jan 13

nginx ssl_reject_handshake set in in helm morriq/kubernetes-raspberry4b#11

Open

senyahnoj mentioned this issue on Jan 17

1.22.5 breaking changes in ingress-nginx canonical/microk8s#2845

Open

kinarashah mentioned this issue on Jan 20

Update nginx v1.1.0 template and bump rke-tools to v0.1.79 rancher/kontainer-driver-metadata#793

Merged

tvories mentioned this issue on Jan 26

Ingress contains invalid word location nextcloud/helm#188

Open

vladciobancai mentioned this issue on Feb 11

CVE-2021-25742: Ingress-nginx digitalocean/Kubernetes-Starter-Kit-Developers#129

Open

tallclair mentioned this issue on Mar 9

KEP-3203: Add Auto-refreshing Official CVE feed kubernetes/enhancements#3204

Merged

dominik-bln mentioned this issue on Apr 4

Usable Helm Chart backstage/backstage#4945

Closed

sathieu commented on Apr 8 Contributor

> rather than opening a new issue, i'm going to use this one to mention an ambiguity in the documentation.
>
> this pr adds documentation for "suggested" strings that should be blocked in `*-snippet` annotations: #7942

Proposed a PR #8446.

dduportal mentioned this issue on Apr 8

fix(jenkinsio) enable again missing security headers jenkins-infra/kubernetes-management#2200

Merged

foxylion commented on Apr 25 Contributor

We were recently notified about this CVE. I see this seems to be fixed in v1.2.0.
Is there anything except from updating to the latest version required to mitgate this type of exploit?

rikatz commented on May 1 Contributor

@foxylion unfortunately not. You can still block the usage of *snippets, but there;s another vulnerability in main ingress object that should be dealt with. What version are you?

I'm closing this issue as it's already fixed in the latest version, and we are planning to deprecate the legacy version so I highly recommend migrating to ingress v1.X

Thanks!
/close

k8s-ci-robot commented on May 1 Contributor

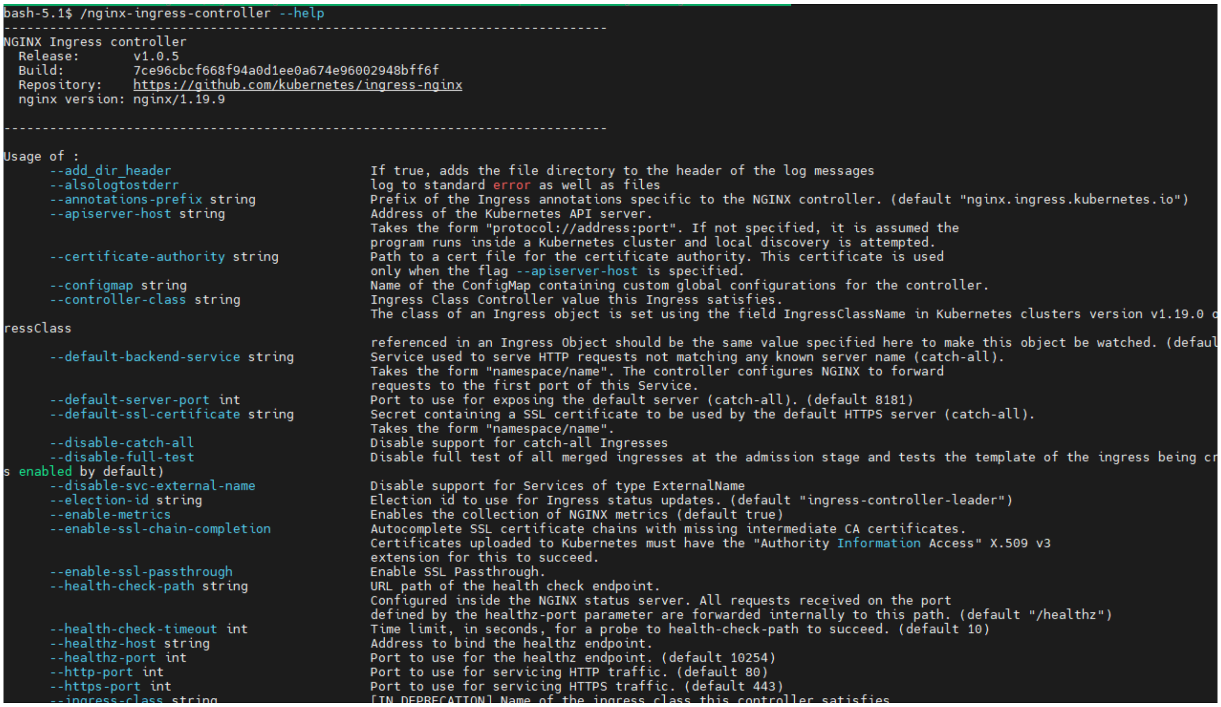
@rikatz: Closing this issue.

▶ Details

k8s-ci-robot closed this as completed on May 1

immanuelfodor commented on May 2

Why is that I do not have an `--enable-snippets` CLI flag on the ingress controller? Is this behind some other feature flag that I need to enable first?



```
bash-5.1$ /nginx-ingress-controller --help
--------------------------------------------------------------------------------
NGINX Ingress controller
  Release:       v1.0.5
  Build:         7ce96cbcf668f94a0d1ee0a674e96002948bff6f
  Repository:    https://github.com/kubernetes/ingress-nginx
  nginx version: nginx/1.19.9

--------------------------------------------------------------------------------

Usage of :
      --add_dir_header                          If true, adds the file directory to the header of the log messages
      --alsologtostderr                         log to standard error as well as files
      --annotations-prefix string               Prefix of the Ingress annotations specific to the NGINX controller. (default "nginx.ingress.kubernetes.io")
      --apiserver-host string                   Address of the Kubernetes API server.
                                                Takes the form "protocol://address:port". If not specified, it is assumed the
                                                program runs inside a Kubernetes cluster and local discovery is attempted.
      --certificate-authority string            Path to a cert file for the certificate authority. This certificate is used
                                                only when the flag --apiserver-host is specified.
      --configmap string                        Name of the ConfigMap containing custom global configurations for the controller.
      --controller-class string                 Ingress Class Controller value this Ingress satisfies.
                                                The class of an Ingress object is set using the field IngressClassName in Kubernetes clusters version v1.19.0 o
ressClass
                                                referenced in an Ingress Object should be the same value specified here to make this object be watched. (defaul
      --default-backend-service string          Service used to serve HTTP requests not matching any known server name (catch-all).
                                                Takes the form "namespace/name". The controller configures NGINX to forward
                                                requests to the first port of this Service.
      --default-server-port int                 Port to use for exposing the default server (catch-all). (default 8181)
      --default-ssl-certificate string          Secret containing a SSL certificate to be used by the default HTTPS server (catch-all).
                                                Takes the form "namespace/name".
      --disable-catch-all                       Disable support for catch-all Ingresses
      --disable-full-test                       Disable full test of all merged ingresses at the admission stage and tests the template of the ingress being cr
s enabled by default)
      --disable-svc-external-name               Disable support for Services of type ExternalName
      --election-id string                      Election id to use for Ingress status updates. (default "ingress-controller-leader")
      --enable-metrics                          Enables the collection of NGINX metrics (default true)
      --enable-ssl-chain-completion             Autocomplete SSL certificate chains with missing intermediate CA certificates.
                                                Certificates uploaded to Kubernetes must have the "Authority Information Access" X.509 v3
                                                extension for this to succeed.
      --enable-ssl-passthrough                  Enable SSL Passthrough.
      --health-check-path string                URL path of the health check endpoint.
                                                Configured inside the NGINX status server. All requests received on the port
                                                defined by the healthz-port parameter are forwarded internally to this path. (default "/healthz")
      --health-check-timeout int                Time limit, in seconds, for a probe to health-check-path to succeed. (default 10)
      --healthz-host string                     Address to bind the healthz endpoint.
      --healthz-port int                        Port to use for the healthz endpoint. (default 10254)
      --http-port int                           Port to use for servicing HTTP traffic. (default 80)
      --https-port int                          Port to use for servicing HTTPS traffic. (default 443)
      --ingress-class string                    [IN DEPRECATION] Name of the ingress class this controller satisfies.
```

MalibuKoKo added a commit to kube-components-stack/helm-charts that referenced this issue on May 31

`CVE-2021-25742 cf. kubernetes/ingress-nginx#7837` ✓ `6103a1a`

This was referenced on Jun 28

[grafana] Fix for cve-2021-25742 is incorrect - not fixable here grafana/helm-charts#1542

Closed

Rule "Prevent ConfigMap security vulnerability (CVE-2021-25742)" is incorrect datreeio/datree#703

Closed

mike-pt mentioned this issue on Aug 18

nginx.ingress.kubernetes.io/server-snippets ignored on deploy #8938

Closed

nikowitt mentioned this issue on Oct 5

URLs with special characters don't work in the presence of URL rewriting #5576

Open

slarwise mentioned this issue on Oct 13

Add annotation for client_header_timeout #9146

Open

Assignees

rikatz

Labels

kind/bug priority/critical-urgent triage/accepted

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

33 participants

and others