New issue

Jump to bottom

Remove hardcoded DNS servers #56

Closed   **fenichelar** wants to merge 1 commit into `home-assistant:master` from `fenichelar:master`

Conversation 54   Commits 1   Checks 0   Files changed 2

fenichelar commented on Sep 24, 2021 • edited

If DNS servers have been manually configured, use the manually configured DNS servers only. Otherwise if DNS servers have been supplied by DHCP, use the DNS servers supplied by DHCP only. Otherwise use Docker Embedded DNS server.

#20

homeassistant commented on Sep 24, 2021

Hi **@fenichelar**,

It seems you haven't yet signed a CLA. Please do so here.

Once you do that we will be able to review and accept this pull request.

Thanks!

homeassistant added **cla-needed** **cla-signed** and removed **cla-needed** labels on Sep 24, 2021

fenichelar commented on Sep 24, 2021   Author

The logic is consistent with the way most systems work:

1. DNS servers can be manually configured and if they are, then only the manually configured DNS servers are used.
2. DNS servers can be configured via DHCP and if they are and no DNS servers have been manually configured, then only the DNS servers configured via DHCP are used.

As an additional fallback, if no DNS servers are manually configured or configured via DHCP, it will use Cloudflare's DNS servers. Most systems don't have functionality like this so DNS requests would just fail, but this may help reduce user issues.

@tescophil **@gjdoornink** **@alexdelprete**

gjdoornink commented on Sep 24, 2021

I do not think it is necessary, or even correct, to remove the Docker Embedded DNS server (dns://127.0.0.11).
Home Assistant can be deployed in multiple ways. Removing the Docker Embedded DNS server might very well impair non standard Docker based deployments. Which might very well, likely even, result in {.servers} being empty.

As far as I interpret the corefile template, it does not treat the Docker Embedded DNS server as a fallback server, but it ensures that either a DHCP supplied DNS server is used or a Docker supplied DNS server.
IMHO this seams quite sensible.

fenichelar commented on Sep 24, 2021   Author

@gjdoornink I updated the logic. Expanded for readability:

```
{{ if .servers }}
  {{ join " " .servers }}
{{ else if .locals }}
  {{ join " " .locals }}
{{ else }}
  dns://127.0.0.11:53 dns://127.0.0.1:5553
{{ end }}
```

Thoughts?

gjdoornink commented on Sep 25, 2021

@fenichelar
To me it seams you have now introduced an issue similar to the one pull request #55 aims to fix.

fenichelar commented on Sep 25, 2021   Author

@gjdoornink How?

gjdoornink commented on Sep 25, 2021

@fenichelar

The hosts in the forward statement form a pool of load balanced DNS resolvers. How the requests are distributed to the resolvers in the pool is determined by the policy. But regardless of which policy is used, requests for local host names will eventually be sent to any of the resolvers in the pool. This means that a certain amount of requests for local host names will inevitably be sent to one of the cloudflare servers and those requests will fail. In pull request #55 I aimed to fix a single use case since I did not feel comfortable overhauling the entire structure without first discussing the feasibility of such a change with the original authors.
If I had the opportunity to do so, my question would be why dns://127.0.0.1:5553 is part of both the forward statement and the fallback statement and not only part of the fallback statement.

But since you are suggesting a more encompassing change ... :-)

Original corefile, without manually configured DNS servers, but with DHCP assigned DNS server:
forward . dns://192.168.0.1 dns://127.0.0.1:5553 {

Your corefile, without manually configured DNS servers and without DHCP assigned DNS server:
forward . dns://127.0.0.11:53 dns://127.0.0.1:5553 {

All requests to dns://127.0.0.1:5553 are forwarded to cloudflare servers which cannot resolve hosts in the local network/docker infrastructure.
dns://192.168.0.1 is a local resolver that can resolve local hosts and, typically, non-local hosts by forwarding requests to other non-local DNS servers.
dns://127.0.0.11:53 is also a local resolver that can resolve local Docker hosts (e.g. containers) and, if so configured through Docker, other local and non-local hosts by forwarding requests to other DNS servers.

fenichelar commented on Sep 25, 2021 • edited Author

@gjdoornink I'm not following what the issue is.

> Your corefile, without manually configured DNS servers and without DHCP assigned DNS server:
> forward . dns://127.0.0.11:53 dns://127.0.0.1:5553 {

Yes. This is the same way it behaves today and the same way it behaves with your PR. If you don't provide DNS servers to Home Assistant that are capable of resolving local domain names, then it won't be able to resolve local domain names. There is nothing we can do about this.

gjdoornink commented on Sep 25, 2021

@fenichelar
What I am trying to express is that the forward statement should not contain DNS resolvers that cannot resolve hosts in the local network, since that inevitably leads to DNS resolve errors and the corresponding problems in Home Assistant.

fenichelar commented on Sep 25, 2021 • edited Author

@gjdoornink

> What I am trying to express is that the forward statement should not contain DNS resolvers that cannot resolve hosts in the local network, since that inevitably leads to DNS resolve errors and the corresponding problems in Home Assistant.

If there are DNS servers that can resolve local domain names, then I agree. But if no DNS servers that can resolve local domain names are known to Home Assistant, then the forward line will contain DNS servers that can't resolve local domains. The only alternative would be to not have any forwarding configured.

Maybe the best way to communicate would be to just let me know what you are thinking for the forwarding line. I'm open to suggestions for my PR. I like your PR but it doesn't solve the problem I am having (I don't want DNS requests to be sent to Cloudflare under any circumstances because the requests are blocked so it just causes noisy logs and slow DNS failures). The issue I am having has been reported by many others as well. So I am trying to solve both of our problems with this PR 😄 .

```
{{ if .servers }}
  {{ join " " .servers }}
{{ else if .locals }}
  {{ join " " .locals }}
{{ else }}
  dns://127.0.0.11:53 dns://127.0.0.1:5553
{{ end }}
```

How would you change the above?

gjdoornink commented on Sep 25, 2021 • edited

@fenichelar

It can indeed be difficult to communicate across an ocean and a language barrier :-)

My statement is that dns://127.0.0.11:53 is also a local resolver just as the DHCP assigned DNS server.

Your pull request tries to resolve a different issue than mine, that is clear.

If I were in your shoes, I would reinstate the following line:
fallback REFUSED,SERVFAIL,NXDOMAIN . dns://127.0.0.1:5553
and replace:
{{ else }}
dns://127.0.0.11:53 dns://127.0.0.1:5553
{{ end }}
with:
{{ else }}
dns://127.0.0.11:53
{{ end }}

But I have not tested this, of course :-)

Kind regards,

fenichelar commented on Sep 25, 2021 Author

@gjdoornink

```
fallback REFUSED,SERVFAIL,NXDOMAIN . dns://127.0.0.1:5553
```

The above line is what is causing my issue. If my DNS server returns any of these errors, then I do not want Home Assistant to try Cloudflare's DNS server. The errors are intentionally returned in many cases (DNSSec failure, domain blocking, domain not found, etc.)

Thoughts on this?

```
{{ if .servers }}
  forward . {{ join " " .servers }} {
    except local.hass.io
    policy sequential
    health_check 1m
  }
{{ else if .locals }}
  forward . {{ join " " .locals }} {
    except local.hass.io
    policy sequential
    health_check 1m
  }
{{ else }}
  forward . dns://127.0.0.11:53 {
    except local.hass.io
    policy sequential
    health_check 1m
  }
  fallback REFUSED,SERVFAIL,NXDOMAIN . dns://127.0.0.1:5553
{{ end }}
```

gjdoornink commented on Sep 25, 2021

@fenichelar
I doubt that what you want is possible by only changing corefile.
IMHO a more sensible approach would be the possibility to configure/override the external fallback DNS servers through some Home Assistant/plugin-dns configuration option, possibly by means of an extension to the 'ha dns' command.
But I must admit that such a project is well outside my own selfish interests (and available time) :-)

fenichelar commented on Sep 25, 2021 Author

@gjdoornink What is wrong with the corefile I provided?

I feel like we are overcomplicating this. Think about how DNS works in ever other device you have (computers, phones, streaming boxes, game consoles, camera, etc.). If DNS servers are manually configured, those DNS servers are used and no other servers. If DNS servers are provided via DHCP, those DNS servers are used and no other servers. Failover is handled by providing multiple DNS servers manually or via DHCP.

In every other device I have, if the configured DNS server returns `NXDOMAIN` , then the device treats this as the domain doesn't exist (which is what it is supposed to do). Why does Home Assistant not trust the DNS servers response and instead try the DNS request again using Cloudflare's DNS servers? Really, why do we want Home Assistant to have this behavior? I can't think of anything else that works this way?

And this has significant drawbacks:

From my Mac:

```
$ dig blahblah.google.com

; <<>> DiG 9.10.6 <<>> blahblah.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 20954
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1432
;; QUESTION SECTION:
;blahblah.google.com.           IN      A

;; AUTHORITY SECTION:
google.com.             57      IN      SOA     ns1.google.com. dns-admin.google.com. 398687860 900 900 1800 60

;; Query time: 3 msec
;; SERVER: 10.1.0.1#53(10.1.0.1)
;; WHEN: Sat Sep 25 12:51:52 EDT 2021
;; MSG SIZE  rcvd: 98
```

From Home Assistant:

```
$ dig blahblah.google.com

; <<>> DiG 9.16.20 <<>> blahblah.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 62546
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 097bd1bc1d16dcf7 (echoed)
;; QUESTION SECTION:
;blahblah.google.com.           IN      A

;; AUTHORITY SECTION:
google.com.             60      IN      SOA     ns1.google.com. dns-admin.google.com. 398687860 900 900 1800 60

;; Query time: 48 msec
;; SERVER: 127.0.0.11#53(127.0.0.11)
;; WHEN: Sat Sep 25 12:51:57 EDT 2021
;; MSG SIZE  rcvd: 140
```

Both my Mac and Home Assistant are connected to the same network, same switch, same DNS server provided via DHCP. Yet Home Assistant took 16 times as long to determine the domain doesn't exist. And this is with Cloudflare not being blocked by my firewall.

And here is an example for DNSSec failure:

Mac:

```
$ dig sigfail.verteiltesysteme.net

; <<>> DiG 9.10.6 <<>> sigfail.verteiltesysteme.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 56067
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1432
;; QUESTION SECTION:
;sigfail.verteiltesysteme.net.  IN      A

;; Query time: 2 msec
;; SERVER: 10.1.0.1#53(10.1.0.1)
;; WHEN: Sat Sep 25 12:51:26 EDT 2021
;; MSG SIZE  rcvd: 57
```

Home Assistant:

```
$ dig sigfail.verteiltesysteme.net

; <<>> DiG 9.16.20 <<>> sigfail.verteiltesysteme.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 27
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c83f04165d38f348 (echoed)
;; QUESTION SECTION:
;sigfail.verteiltesysteme.net.  IN      A

;; Query time: 280 msec
;; SERVER: 127.0.0.11#53(127.0.0.11)
;; WHEN: Sat Sep 25 12:51:14 EDT 2021
;; MSG SIZE  rcvd: 69
```

Home Assistant was 140 times slower.

I only left Cloudflare in the corefile because I wanted to change as little as possible and solve the issue. But I really don't think any DNS server should be hardcoded like that. What other device has hardcoded DNS servers? Personally, I think the config should be:

```
.:53 {
    log {{ if not .debug }}{
        class error
    }{{ end }}
    errors
    loop
    {{ if .debug }}debug{{ end }}
    hosts /config/hosts {
        fallthrough
    }
    template ANY AAAA local.hass.io hassio {
        rcode NOERROR
    }
    mdns
    forward . {{ if .servers }}{{ join " " .servers }}{{ else if .locals }}{{ join " " .locals }}{{ else }}dns://127.0.0.11:53{{ end }} {
        except local.hass.io
        policy sequential
        health_check 1m
    }
    cache 600
}
```

Which makes Home Assistant behave just like every other device does. I don't understand why we need hardcoded DNS servers.

Remove hardcoded DNS servers ✓ efa0011

fenichelar changed the title ~~Improve DNS server selection~~ **Remove hardcoded DNS servers** on Sep 25, 2021

Zixim commented on Sep 26, 2021 • edited

what would happen if :
user is running Home Assistant OS, user hasn't set a DNS server, user isn't using DHCP ?

Requests would go to the Docker embedded server (dns://127.0.0.11:53), right ?
Is the docker embedded dns even active in Home Assistant OS ?

I think this might be the reason why the core devs added cloudflare servers.

Any solution to this dns mess shouldn't remove the ultimate fallback, even though in any half-decent network setup this shouldn't ever happen.

fenichelar commented on Sep 26, 2021 • edited Author

@Zixim

> user is running Home Assistant OS, user hasn't set a DNS server, user isn't using DHCP ?
> Requests would go to the Docker embedded server (dns://127.0.0.11:53), right ?

Yes

> Is the docker embedded dns even active in Home Assistant OS ?

I'm not sure but the way it currently works if no DNS servers are configured manually or with DHCP is that requests are sent to the Docker Embedded DNS server and then Cloudflare. I just removed Cloudflare but kept the rest of the logic essentially the same.

> Any solution to this dns mess shouldn't remove the ultimate fallback, even though in any half-decent network setup this shouldn't ever happen.

Can you explain why you think Home Assistant should have a failover DNS server if one is not provided manually or through DHCP? I think DNS requests should fail in this scenario. Does you computer, phone, streaming box, game console, printer, or any other network device have a failover DNS server? Mine don't. Home Assistant is the only device in my network that has a hard coded failover DNS server as far as I know. I truly don't understand why Home Assistant should be any different than every other device. I know making Home Assistant "just work" is an important goal, but Apple has the same goal with their products and neither Macs nor iOS devices have a hard coded failover DNS servers. If DNS doesn't work in your network, none of your devices are going to work. I think Home Assistant not working in this scenario is not only acceptable, it is appropriate.

That being said, I am certainly happy to contribute on a PR that makes failing over to Cloudflare's DNS servers optional. But I am not sure I am familiar enough with supervisor (where the config option would be managed) to try and tackle this on my own.

Zixim commented on Sep 26, 2021

> Can you explain why you think Home Assistant should have a failover DNS server if one is not provided manually or through DHCP?

It *shouldn't* have a failover. I fully agree with your reasoning. My only concern is that this PR might get refused/ignored because it is contrary to what core devs have decided.

> That being said, I am certainly happy to contribute on a PR that makes failing over to Cloudflare's DNS servers optional. But I am not sure I am familiar enough with supervisor (where the config option would be managed) to try and tackle this on my own.

Couldn't your logic/PR be extended to add the failover into the corefile ?
Much like this part (for port 5553) in the corefile :

```
.:5553 {
    log {{ if not .debug }}{
        class error
    }{{ end }}
    errors
    {{ if .debug }}debug{{ end }}
    forward . tls://1.1.1.1 tls://1.0.0.1 {
        tls_servername cloudflare-dns.com
        except local.hass.io
        health_check 5m
    }
    cache 600
}
```

tescophil commented on Sep 26, 2021

For me, Core DNS was always a solution looking for a problem, DNS has worked the same way for decades and does not need 'fixing''

HA should use DNS servers configured by DHCP or the user directly. If none are set, or the ones that are set fail, then HA should just fail. Adding a fallback is just moving a network problem elsewhere and making it more difficult to trace and resolve.

Personally I think hard coded DNS of any kind is a breach of the users privacy and trust and should be removed.

If the core devs believe that there should be a fallback (as I believe they do, having had four requests to change this over the last 12 months rejected), then:

Firstly it should just use plain port 53 unencrypted DNS, not DoT. This is because port 53 DNS requests can easily be redirected to a local/alternate DNS server at the network firewall, this cannot be done with DoT requests as the SSL cert of a local DoT server will not match the requested domain.

Secondly, the fallback server should only be contacted on SERVFAIL, because NXDOMAIN and REFUSED are not errors.

Thirdly, currently if the fallback is blocked, a cascading failure mode occurs. HA gets stuck in a loop sending millions of DNS request to the fallback to the point where all other components become unresponsive. (the fallback for the fallback is the fallback). Clearly if the fallback fails, it's should not be contacted again.

Zixim commented on Sep 26, 2021

perhaps trying a parallel PR which does nothing more than removing NXDOMAIN and REFUSED from the corefile ?

fenichelar commented on Sep 26, 2021 • edited ▾ Author

@tescophil @Zixim

> If the core devs believe that there should be a fallback (as I believe they do, having had four requests to change this over the last 12 months rejected), then:

Maybe they do, maybe they don't. At least some of the requests were hostile and just turned into unproductive name calling. I also haven't seen any PRs. Submitting an issue asking a maintainer to do something (particularly in an aggressive tone) and submitting a PR are very different. I'm not trying to point fingers at anyone, I am just saying that I think a productive conversation with the maintainers should be had about this PR before we come to the conclusion that this PR or a variation of it will not be accepted 😄.

> Secondly, the fallback server should only be contacted on SERVFAIL, because NXDOMAIN and REFUSED are not errors.

> perhaps trying a parallel PR which does nothing more than removing NXDOMAIN and REFUSED from the corefile ?

`SERVFAIL` is an expected response when DNSSEC chain verification fails. It should not result in the fallback servers being used either. This is why I completed removed the hardcoded failover for all responses and not just for `NXDOMAIN` and `REFUSED` responses.

Below is an **incomplete** list of potential paths forward:

1. Completely remove the hard coded DNS fallback servers (what this PR currently does).
2. Provide an option to disable the DNS fallback servers (wrap the changes made in this PR behind a new configuration flag). I can't imagine anyone would be against this approach. The only reason I haven't just done this is because I am not familiar enough with the Home Assistant code base to do it. Where are all the places the configuration flag has to be added and documented? But if someone can give me some high level directions, I can start working in it.
3. Only use DNS fallback servers is DNS servers were provided by DHCP. Basic users that use DHCP would have fallback DNS servers. More advanced users could manually set their DNS servers to disable the fallback DNS servers. This is basically option 2 above but uses the manually configured list of DNS servers being present as the configuration flag. Meaning we don't need to add a new configuration flag. I can easily implement this PR if that is what is decided.

Hopefully one of the maintainers can response providing some directions on how they want to proceed. I'm optimistic we can figure this out 😄 .

❤️ 2

alexdelprete commented on Sep 26, 2021

In 25y of IT experience as a system and network administrator, I never found a server software, or an enterprise sw, or an automation sw in this case, that extends the network configuration of the user/environment in which it runs, by auto-provisioning such an important thing as a FALLBACK DNS server. It's a bad design and practice. If the environment is not properly configured, it is CORRECT that the sw doesn't work, it should simply throw an error. Network configuration should never be decided by an app, it is up to the user/admin to make those choices. If they don't have the skills to decide, it means they're using standard DHCP, and that's it.

Can you simply imagine a WEB SERVER "fallbacking" to a fixed external DNS server? Gosh...it would be considered a serious security violation.

I really can't understand what was the idea behind the fallback enry in that configuration. If some dev could transparently explain what was the logic behind that decision, we could maybe understand better what most people consider a bad implementation.

If the network environment is not properly configured, it is absolutely correct that HA doesn't work. HA should throw an error when it has DNS issues, like properly designed apps/frameworks/enterprise sw do, that's it.

Furthermore, those benchmarks I read about DNS performance in HA would explain a lot of timeout/connection errors we see in HA logs. DNS is a critical service, it should be reliable, fast, efficient. Actual implementation is not at the level of quality we expect from HA.

All this discussion about this DNS/networking architecture is almost surreal...

fenichelar commented on Sep 26, 2021 Author

@alexdelprete

> If the network environment is not properly configured, it is absolutely correct that HA doesn't work. HA should throw an error when it has DNS issues, like properly designed apps/frameworks/enterprise sw do, that's it.

I think you are bringing up a good point. We need to make sure that Home Assistant logs a clear error message if no DNS servers have been provided via DHCP or manually. It would also be good if Home Assistant logged clear error messages if the health checks to the DNS servers fail. I will dig into what the logs look like today under these scenarios and see if any improvements can be made.

billgertz commented on Sep 27, 2021

@fenichelar

Agree that this is a good point - I ran into this as I was investigating issues caused by robots aggressively scanning intentionally exposed web servers. Doing due diligence and troubleshooting when I ran into this CloudFlare DoH problem. (Yes it's a problem, as I had made no conscious choice as the Security Architect for this functional requirement). Had to do a lot of digging to realize what was going on. Disturbed as this is neither a configurable item nor is it mentioned in the documentation.

This *must* be documented so that the requirement is known and security administrators will expect this in their logs.

I would hope the developers would allow an option to, um, opt out. If not that then at least *clearly document* what is going on.

tescophil commented on Sep 27, 2021

I have had discussions with the devs in the past on this subject, and it boils down to this: They believe that more 'issues' will be raised, that they then have to deal with, if a fallback DNS service is removed.

Implementing a fallback DNS is more common than you would think, and is done by small and large organisations alike. eg. All Google devices (Google Home Hub, Google Speakers Nest Hubs, Chromecast, Android Phones, Nest Thermostats etc.) are all hard coded to use Googles public DNS servers, some as a fallback, and some just ignore any local DNS settings all together.

The difference is when, for example, you block access to Google public DNS on your local network, these devices carry on working without issues using the locally configured DNS. HA does not. Block access to CloudFlair on your local network and HA goes into meltdown, sending out millions of requests until eventually the system becomes overloaded and fails.

IMHO removing this all together would be the best option, but if it has to stay, then at least fix the 'meltdown' issue if access to the fallback is blocked, and/or move away from DoT and use basic DNS on Port 53 which can be easily redirected to a local service.

alexdelprete commented on Sep 27, 2021 • edited

> They believe that more 'issues' will be raised, that they then have to deal with, if a fallback DNS service is removed.

If that is true, they should implement a configurable option to disable it, leaving it as default (horrible decision). You can't create problems to many people because you "think" you might have more issues. Let these things be configurable, hardcoding these kind of things is really a bad choice, especially for an open-source solution.

> Implementing a fallback DNS is more common than you would think, and is done by small and large organisations alike

Sorry but I mentioned enteprise software and apps, not hardware devices designed explicitly to work with cloud. And even in that case, it is bad practice and design. My firewall disables all kind of dns traffic from LAN to WAN. I enforce usage of my local DNS server, had to do it exactly because many bad designed firmware in devices behave like that. But you'll never find a commercial enterprise app with a fallback dns not configurable, because it's an awful design choice.

> Block access to CloudFlair on your local network and HA goes into meltdown

My firewall/router (OPNsense) has been configured to block all DNS traffic to the internet, except from the DNS server itself, that that runs on the firewall. HA doesn't meltdown, it works. But I stumbled across the DNS problem because sometimes, only some times, it starts wanting to use the CloudFlare servers, and it can't, so it starts not working. But 95% of the time it works fine. I didn't notice the "millions of queries" issue. What I noticed is that generally, HA is really SLOW in resolving hostnames, because of the awful dns design/implementation.

3 hidden items

Load more...

pvizeli closed this on Sep 28, 2021

pvizeli mentioned this pull request on Sep 28, 2021

Prevent intermittent DNS failures when DNS server in local network is used. #55

Closed

Zixim commented on Sep 28, 2021

> Are there any core maintainers seeing this that can help provide some direction on what we are trying to accomplish with failover?

core maintainer :

> That will not solve the main issue which we solved with this.

May we know what is/was the main issue that you solved with this ?

alexdelprete commented on Sep 28, 2021 • edited

> That will not solve the main issue which we solved with this

May we kindly ask to know what is the "main issue" you tried to solve? Maybe there are better ways to solve it, without creating other issues.

> - Open a PR to Supervisor and add an option to disable fallback which marks your system as unsupported and extend it to the API + tests
> - Create an PR to the developer documentation for having it there
> - Create an PR to CLI repository for using that options
> - Create an PR to this repository to get this options in place

So, if we suggest to disable the fallback, explaining why, you ask for all kinds of PR and options to be implemented, but when you implemented the fallback, you didn't provide everything needed to enable/disable it. May I kindly ask why you didn't put that option in first place when you decided for the hard-coded fallback?

Last question: why would a system with a disabled fallback option be considered "unsupported"?

I really hope HA is still an open-source project, because it seems like when users have different opinions vs the official devs, there are too many imposed obstacles to overcome. Collaboration should be the way to pursue, not the opposite.

The DNS implementation of HA is clearly badly designed and inefficient (**@fenichelar** benchmarked it, it is amazingly slow and buggy). We're asking to improve it because we love HA and would like the product to be better than it currently is. I look forward to devs helping us help them realize that there's much to improve regarding DNS. But that doesn't seem the case...

You guys, with all other contributors, made an incredible product, it's a shame that a fundamental functionality of the system is not at the same general level, but the worst thing is you don't even want to discuss about it.

BTW: you also closed #55 which has nothing to do with #56, it tackles a totally separate issue (yes, DNS has several issues).

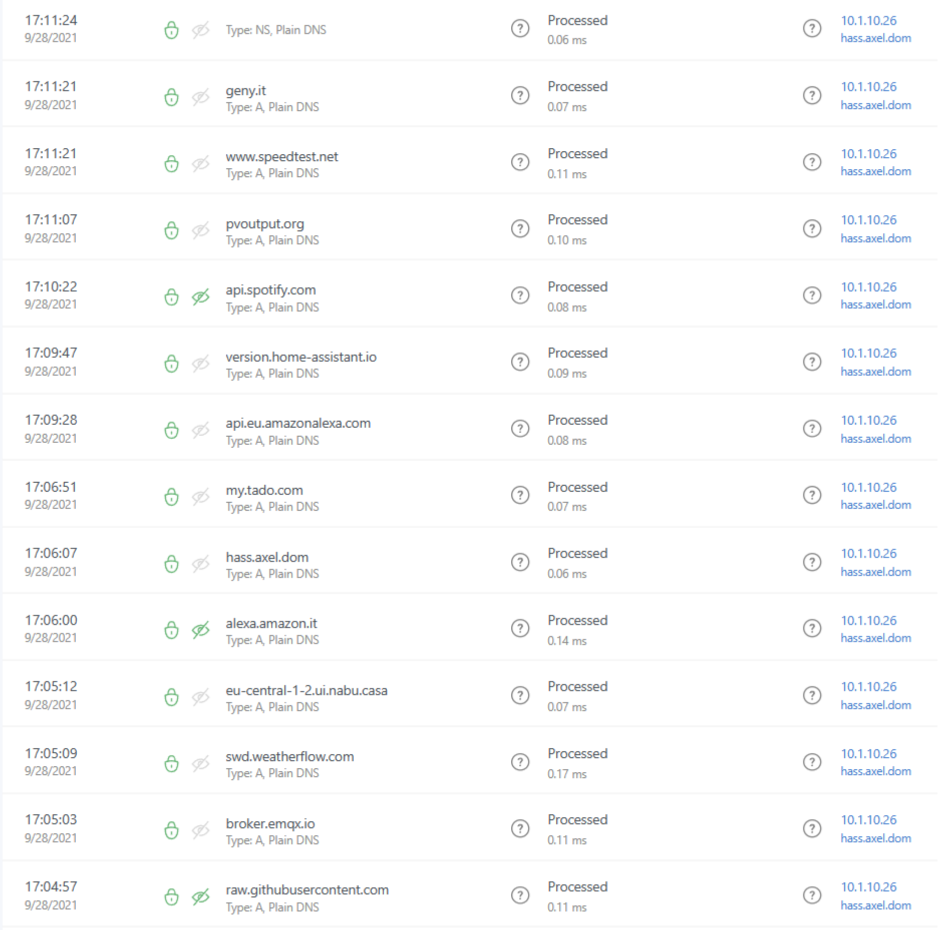
👍 2

alexdelprete commented on Sep 28, 2021 • edited

> So, when I block access to CloudFlair on my network, after a few hours I see this:

I don't have that kind of behaviour, but sometimes I have the problem that for some strange reasons HA decides to use cloudflare also for local resolution, and it obviously fails because any DNS lookup to the internet is blocked. See #55.

This is what I see in the dns logs, with internet dns traffic blocked (default rule in my firewall):



17:11:24 9/28/2021	Type: NS, Plain DNS	Processed 0.06 ms	10.1.10.26 hass.axel.dom
17:11:21 9/28/2021	geny.it Type: A, Plain DNS	Processed 0.07 ms	10.1.10.26 hass.axel.dom
17:11:21 9/28/2021	www.speedtest.net Type: A, Plain DNS	Processed 0.11 ms	10.1.10.26 hass.axel.dom
17:11:07 9/28/2021	pvoutput.org Type: A, Plain DNS	Processed 0.10 ms	10.1.10.26 hass.axel.dom
17:10:22 9/28/2021	api.spotify.com Type: A, Plain DNS	Processed 0.08 ms	10.1.10.26 hass.axel.dom
17:09:47 9/28/2021	version.home-assistant.io Type: A, Plain DNS	Processed 0.09 ms	10.1.10.26 hass.axel.dom
17:09:28 9/28/2021	api.eu.amazonalexa.com Type: A, Plain DNS	Processed 0.08 ms	10.1.10.26 hass.axel.dom
17:06:51 9/28/2021	my.tado.com Type: A, Plain DNS	Processed 0.07 ms	10.1.10.26 hass.axel.dom
17:06:07 9/28/2021	hass.axel.dom Type: A, Plain DNS	Processed 0.06 ms	10.1.10.26 hass.axel.dom
17:06:00 9/28/2021	alexa.amazon.it Type: A, Plain DNS	Processed 0.14 ms	10.1.10.26 hass.axel.dom
17:05:12 9/28/2021	eu-central-1-2.ui.nabu.casa Type: A, Plain DNS	Processed 0.07 ms	10.1.10.26 hass.axel.dom
17:05:09 9/28/2021	swd.weatherflow.com Type: A, Plain DNS	Processed 0.17 ms	10.1.10.26 hass.axel.dom
17:05:03 9/28/2021	broker.emqx.io Type: A, Plain DNS	Processed 0.11 ms	10.1.10.26 hass.axel.dom
17:04:57 9/28/2021	raw.githubusercontent.com Type: A, Plain DNS	Processed 0.11 ms	10.1.10.26 hass.axel.dom

Would love to reproduce the same issue you have...

billgertz mentioned this pull request on Sep 28, 2021

Document Fallback Behaviour and Functional Requirements home-assistant/home-assistant.io#19511

Closed

billgertz commented on Sep 28, 2021 • edited

As you can see from the reference blurb above I'm asking that his behavior and associated functional requirements be documented. As it stands now this is a hidden feature and causing some heartburn and performance issues for some of us.

Let's see if we can get this documented.

👀 1

alexdelprete commented on Sep 28, 2021

> Let's see if we can get this documented.

I agree that documenting it would be more appropriate, but it wouldn't solve the fundamental issue.

At least we would know the reasons behind the decision to enforce a harcoded fallback without opt-out option. Will be very interesting reading it.

fenichelar commented on Sep 28, 2021 Author

@alexdelprete I am able to reproduce the **@tescophil** described. Try running `dig blahblahblah.google.com` on HA. You should get an `NXDOMAIN` response from your DNS server, which will cause HA to try Cloudflare. You may also want to try `dig sigfail.verteiltesysteme.net` . Does your local DNS server support DNSSEC?

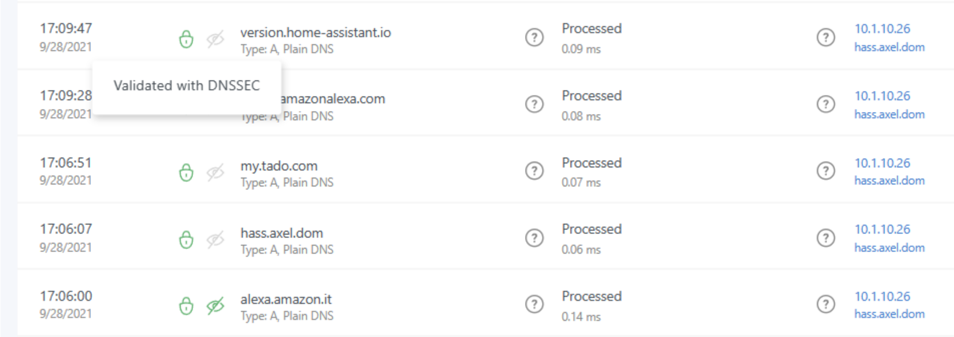
fenichelar commented on Sep 28, 2021 Author

@pvizeli Would a PR that changes the Cloudflare DNS servers to UDP instead of TLS be accepted? This would allow the DNS queries to be redirected to a different server while still providing failover like it does today.

alexdelprete commented on Sep 28, 2021

> Does your local DNS server support DNSSEC?

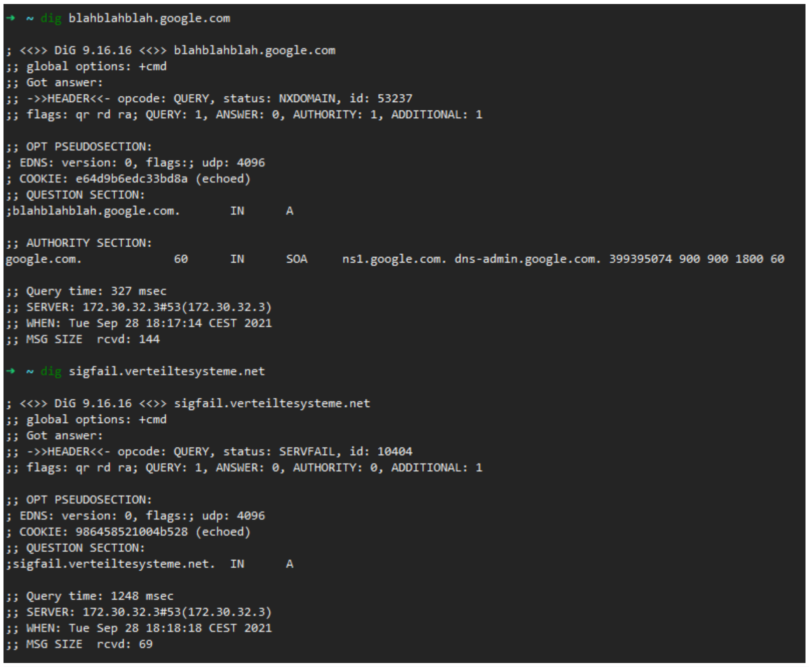
yes, all those green locks you see mean "DNSSEC validated" query. That's AdGuardHome on OPNsense, that acts as LAN DNS server, it then forwards all requests to unbound on OPNsense that is properly configured for DNSSEC, etc. I use AdGuard to filter, and unbound to do the real requests.



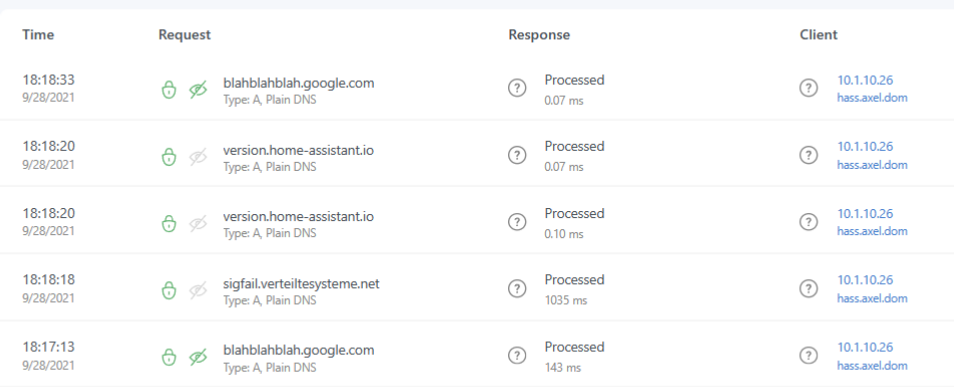
Validated with DNSSEC

17:09:47 9/28/2021	version.home-assistant.io Type: A, Plain DNS	Processed 0.09 ms	10.1.10.26 hass.axel.dom
17:09:28 9/28/2021	amazonalexa.com Type: A, Plain DNS	Processed 0.08 ms	10.1.10.26 hass.axel.dom
17:06:51 9/28/2021	my.tado.com Type: A, Plain DNS	Processed 0.07 ms	10.1.10.26 hass.axel.dom
17:06:07 9/28/2021	hass.axel.dom Type: A, Plain DNS	Processed 0.06 ms	10.1.10.26 hass.axel.dom
17:06:00 9/28/2021	alexa.amazon.it Type: A, Plain DNS	Processed 0.14 ms	10.1.10.26 hass.axel.dom

These are the DIG tests:



```
➜ ~ dig blahblahblah.google.com

; <<>> DiG 9.16.16 <<>> blahblahblah.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 53237
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: e64d9b6edc33bd8a (echoed)
;; QUESTION SECTION:
;blahblahblah.google.com.       IN      A

;; AUTHORITY SECTION:
google.com.             60      IN      SOA     ns1.google.com. dns-admin.google.com. 399395074 900 900 1800 60

;; Query time: 327 msec
;; SERVER: 172.30.32.3#53(172.30.32.3)
;; WHEN: Tue Sep 28 18:17:14 CEST 2021
;; MSG SIZE  rcvd: 144

➜ ~ dig sigfail.verteiltesysteme.net

; <<>> DiG 9.16.16 <<>> sigfail.verteiltesysteme.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 10404
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 986458521004b528 (echoed)
;; QUESTION SECTION:
;sigfail.verteiltesysteme.net.  IN      A

;; Query time: 1248 msec
;; SERVER: 172.30.32.3#53(172.30.32.3)
;; WHEN: Tue Sep 28 18:18:18 CEST 2021
;; MSG SIZE  rcvd: 69
```



Time	Request	Response	Client
18:18:33 9/28/2021	blahblahblah.google.com Type: A, Plain DNS	Processed 0.07 ms	10.1.10.26 hass.axel.dom
18:18:20 9/28/2021	version.home-assistant.io Type: A, Plain DNS	Processed 0.07 ms	10.1.10.26 hass.axel.dom
18:18:20 9/28/2021	version.home-assistant.io Type: A, Plain DNS	Processed 0.10 ms	10.1.10.26 hass.axel.dom
18:18:18 9/28/2021	sigfail.verteiltesysteme.net Type: A, Plain DNS	Processed 1035 ms	10.1.10.26 hass.axel.dom
18:17:13 9/28/2021	blahblahblah.google.com Type: A, Plain DNS	Processed 143 ms	10.1.10.26 hass.axel.dom

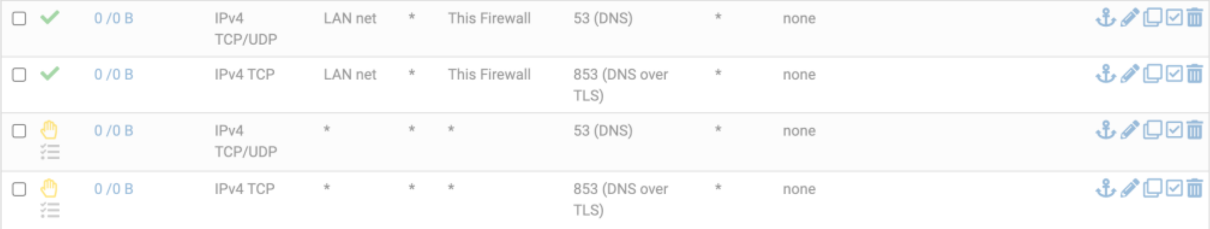
I don't see all those queries...

fenichelar commented on Sep 28, 2021 Author

@alexdelprete You won't see the queries because they use TLS. Setup packet logging in your pfSense firewall rules that block TCP port 853 and you should see all of the traffic.

fenichelar commented on Sep 28, 2021 • edited Author

@alexdelprete



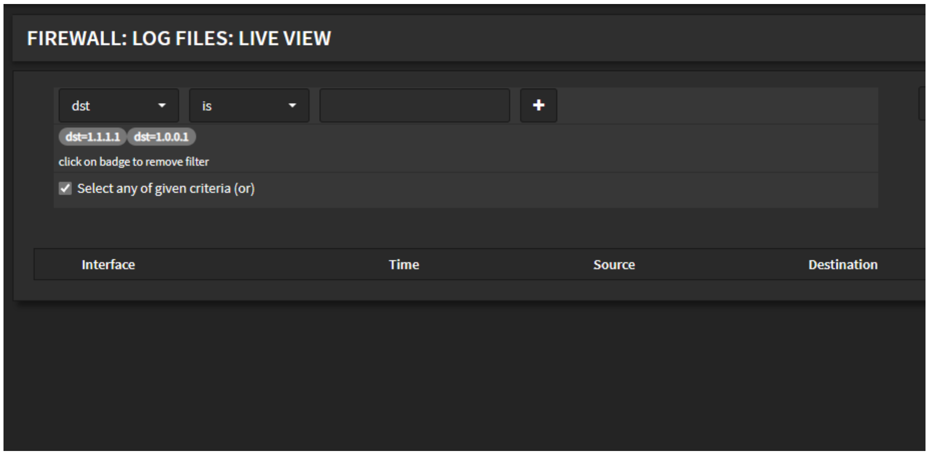
□	✓	0 /0 B	IPv4 TCP/UDP	LAN net	*	This Firewall	53 (DNS)	*	none	
□	✓	0 /0 B	IPv4 TCP	LAN net	*	This Firewall	853 (DNS over TLS)	*	none	
□	✋	0 /0 B	IPv4 TCP/UDP	*	*	*	53 (DNS)	*	none	
□	✋	0 /0 B	IPv4 TCP	*	*	*	853 (DNS over TLS)	*	none	

I had to disable the rules because it broke HA, but that is how you can see the traffic HA produces.

alexdelprete commented on Sep 28, 2021 • edited

> You won't see the queries because they use TLS. Setup packet logging in your pfSense firewall rules that block TCP port 853 and you should see all of the traffic.

I know, I checked for 1.1.1.1 and 1.0.0.1 in the logs, I see no records after the dig.



FIREWALL: LOG FILES: LIVE VIEW

dst ▾ is ▾ +

dst=1.1.1.1 dst=1.0.0.1

click on badge to remove filter

☑ Select any of given criteria (or)

Interface	Time	Source	Destination

frenck commented on Sep 28, 2021 Member

This PR has been closed and possible/acceptable guidance has been provided. Meanwhile the discussion goes on and is no longer about the code provided in this PR but slowly becoming a more generic discussion. Please use our community forum or Discord chat for these purposes.

Thanks!

../Frenck

alexdelprete commented on Sep 28, 2021

> Please use our community forum or Discord chat for these purposes.

Frenck, what's the appropriate discord channel to discuss DNS?

frenck commented on Sep 28, 2021 Member

depends on what you'd like to discuss, probably helps to ask around on Discord for that matter, not GitHub.

fenichelar commented on Sep 28, 2021 • edited Author

@frenck

> Meanwhile the discussion goes on and is no longer about the code provided in this PR but slowly becoming a more generic discussion.

I am very much still focused on updating this PR.

We are hoping to get an answer to this question: #56 (comment)

> Are there any core maintainers seeing this that can help provide some direction on what we are trying to accomplish with failover?

Can you please help us understand? The current failover implementation causes a lot of problems. For example, if internet connectivity is lost, HA slows to a halt on low powered devices because of the constant attempts to establish a TLS connection to Cloudflare. We are trying to understand what the goal of failover is so that this PR can be adjusted accordinly.

Also, can you help with this?

> Would a PR that changes the Cloudflare DNS servers to UDP instead of TLS be accepted? This would allow the DNS queries to be redirected to a different server while still providing failover like it does today.

frenck commented on Sep 28, 2021 Member

> I am very much still focused on updating this PR.

> Also, can you help with this?

This PR is closed.

👎 2

billgertz commented on Sep 28, 2021

Thanks for the help :-/

fenichelar commented on Sep 28, 2021 Author

@frenck We are all just trying to solve the problems we are facing. Two different PRs were implemented and both closed. When we try to discuss what PR would be accepted, we just get shot down.

We appreciate the small amount of guidance received. But we were told to go do a bunch of software development if we want to solve this problem but when we try to get more guidance, we can't get any information.

If you don't want to have this conversation on GitHub, then can you provide the specific location you want to have it?

I understand there has been some hostility towards HA maintainers about this issue in the past. But that was coming from different people, don't punish us for their behavior.

👍 1

frenck commented on Sep 28, 2021 Member

> **@frenck** We are all just trying to solve the problems we are facing.

I'm trying to stop notifications and discussions on a closed PR; especially it's no longer discussing the actual code inside this PR. It generates tons of notifications that aren't actionable to any of us.

> When we try to discuss what PR would be accepted, we just get shot down.

See: #56 (review)

> If you don't want to have this conversation on GitHub, then can you provide the specific location you want to have it?

See: #56 (comment)

> I understand there has been some hostility towards HA maintainers about this issue in the past. But that was coming from different people, don't punish us for their behavior.

We are not punishing, we don't consider it an issue, but are open for a change as written above.

fenichelar commented on Sep 28, 2021 Author

> I'm trying to stop notifications and discussions on a closed PR; especially it's no longer discussing the actual code inside this PR

I am trying to discuss the code in this PR. I want to update the code so that it is accepted.

> See: #56 (review)

Can you add any more details to this comment? Specifically:

> That will not solve the main issue which we solved with this. If you want this then you need to do the following:

What issue?

> Open a PR to Supervisor and add an option to disable fallback which marks your system as unsupported and extend it to the API + tests

What should be config flag be called? What should its type be?

> Create an PR to the developer documentation for having it there

Okay

> Create an PR to CLI repository for using that options

Okay

> Create an PR to this repository to get this options in place

What specifically should this flag do? Remove the Cloudflare DNS servers from the forward line, failover line, or both?

> See: #56 (comment)

Does that mean Discord is the right place?

> We are not punishing, we don't consider it an issue, but are open for a change as written above.

Are you saying that when Home Assistant losses internet connectivity the constant flood of TCP connection attempts slowing down HA is expected and therefore isn't an issue, is an unrelated issue, or is not something you have been able to reproduce?

alexdelprete commented on Sep 28, 2021 • edited

> I'm trying to stop notifications and discussions on a closed PR

You can reopen it. If somebody decided to close it because there's no will for a serious discussion, doesn't mean it's a valid reason to close a PR and then use the closing as a reason not to discuss. Looks like a good method to you, but it's clear it's a method to avoid discussions.

> we don't consider it an issue

That's the real issue that is neither logic nor understandable: can you, Pascal, or someone else explain why it's not an issue for you and it's an issue for many other people? We provided several arguments for which it IS an issue, but haven't read any serious factual argument from the ones who close PRs because they don't consider this an issue.

You might as well close any possibility to open a PR for the DNS plugin, would avoid these discussions entirely. :)

> but are open for a change as written above.

That is not true: Pascal said that even following those guidelines, the systems that opt-out would be considered "unsupported". The reason for this? Not discussed, undisclosed.

❤️ 2

alexdelprete commented on Sep 28, 2021

> Are you saying that when Home Assistant losses internet connectivity the constant flood of TCP connection attempts slowing down HA is expected and therefore isn't an issue, is an unrelated issue, or is not something you have been able to reproduce?

It's not an issue for them, full stop. They don't care if it's an issue for other users. The PR is closed, so you can't even discuss, because...it's closed, in agreement with who opened it. ;)

👍 2

This comment has been minimized. Sign in to view

Zixim commented on Sep 28, 2021

also, stop mentioning Them, you'll get muzzled.
Ivory tower developing & gatekeeping FTW...

frenck commented on Sep 28, 2021 Member

I'm happy to discuss anything, all I asked was to stop discussing this on a closed PR.

Yet, seems like that is ignored completely and additionally some snarky comments need to be made, which is really sad and doesn't motivate me at all...

Locking down this PR as resolved/heated/off-topic.

home-assistant locked as **off-topic** and limited conversation to collaborators on Sep 28, 2021

balloob commented on Sep 28, 2021 Member

We make the decisions we do because with Home Assistant OS we are offering a solutions to users that works, for users that want to focus on about home automation, not system administration.

We used to get a ton of issues with people incorrectly configuring their DNS and then Home Assistant stops working. All those issues disappeared when we did this.

If you're interested in doing DNS stuff, probably Home Assistant OS is not for you. Consider using a Supervised or Container installation.

Now Pascal did mention a way the forwarding can be resolved, but that people in this PR consider too much work nor do commenters agree with it being marked as unsupported. You can't have it all. Stray off the easy path, accept the consequences. If you want it to remain supported, it means it becomes *our* problem, because we are supporting it then. We are not interested in taking up any extra work that a) slows down any future improvements and b) benefits only a minority of our users.

And as usual, any topic about DNS ends up with insults towards us, so it's closed.

We are not interested in further discussing this topic.

Reviewers

pvizeli

Assignees

No one assigned

Labels

cla-signed

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

11 participants