GSoC 2019 Proposal
Organization: CNCF (coredns)

Student Info:

- **Name:** Palash Nigam
- **GitHub username:** @palash25
- **Email:** npalash25@gmail.com
- **Location:** Lucknow, U.P. , India.
- **Time Zone:** UTC+05:30
- **GSoC blog RSS feed URL:** @npalash25 (Medium account)

Contributions to coredns:

Issue	Description	PR	Status
#1407	plugin/secondary: add metrics	#2550	Under review
-	fix link to whoami plugin page	#139	Merged
#1520	plugin/rewrite: Add metrics	#2767	Under review
#2695	Dedup code between grpc and forward plugin	#2771	WIP

Project Info:

Title	Source IP based block/allow mechanism

Mentors	Yong Tang @yongtang

Abstract:

CoreDNS is a DNS server written in Go and built on top of of the caddy web server that provides a pluggable architecture to be able to build and customize DNS solutions according to a particular use case.

Why is this project needed?
CoreDNS provides a rich set of plugins but none of them offer protection to the DNS server from being attacked by a malicious sources. Because of this there is a need to develop a firewall like plugin that once configured in the Corefile will only allow requests from certain source IPs or CIDR blocks to pass and the rest will be rejected

Project Deliverables and Goals:

This project aims to develop a plugin for coredns that will be able to block/allow requests based on the source IPs.

The plugin can be further enhanced by adding metrics for every request blocked and allowed.

Add unit tests for the plugin that will test the functionality of the plugin in various test cases such as blocking/allowing individual IPs or CIDR blocks.

The final goal of the project would be to make the plugin ready to be merged with the coredns codebase as an internal plugin

Implementation Details:

CoreDNS provides a plugable architecture which makes it easy to write and integrate both internal and external plugins for it. The project will start with writing a basic plugin that can read various source IPs (strings) in a corefile. For example:

```
.:1053 {
        firewall {
           allow "127.0.0.1"
           block “192.168.0.1”
        }
}
```

During the first phase the work will be centered around building a basic plugin that can read individual IPs

Write a basic plugin that will parse the list of IPs mentioned in the firewall block and store it as a map of string IPs and actions (block or allow), key-value pairs.

```
func setup(c *caddy.Controller) error {
   // parse function iterates over the Corefile and collects all the IPs and actions in a map
   fw, _ := parse(c)

   dnsserver.GetConfig(c).AddPlugin(func(next plugin.Handler) plugin.Handler {
      fw.Next = next
      return fw
   })

   return nil
}
```

The config map would look something like this

```
var parsedFirewallConfig = map[string]string{
   "127.0.0.1": "allow",
   "192.168.0.1": "block",
   // and so on ...
}
```

The serverDNS method will try to match the request IP with one of the parsed IPs and return with either ***"NXDOMAIN"*** or ***"NOERROR"*** and an appropriate message depending on whether the action was block or allow respectively

```
func (p Demo) ServeDNS(ctx context.Context, w dns.ResponseWriter, r *dns.Msg) (int, error) {
    state := request.Request{W: w, Req: r}
    qname := state.Name()

    var reply string

    val, ok := parsedFirewallConfig[state.IP()]
    // if IP not found in config file then block it by default
    if !ok {
        val = "block"
    }
    status := dns.RcodeNameError

    if val == "allow" {
        status = dns.RcodeSuccess
        reply = "Some success message"
    } else {
        reply = "Some failure message"
    }

    // Write the dns record
    // Set the status and message
}
```

Further the plugin will be modified to read CIDR blocks from the Corefile and to be able to process them along with individual IPs too. Further we could also enable the plugin to read from IP whiltelists or blacklists files in order to provide some basic mitigation against DNS amplification attacks. E.g.

```
.:1053 {
        firewall {
           allow 192.168.0.0/23
           block “127.0.0.1”
           whitelist “/etc/ip_whitelist.txt”
           blacklist “/etc/ip_blacklist.txt”
        }
}
```

Another enhancement that could be made to the plugin is the addition of metrics to record the number of requests blocked and allowed. These metrics will be counter vectors the will be incremented every time a request gets blocked or allowed

```
BlockedRequestsCount = prometheus.NewCounterVec(prometheus.CounterOpts{
    Namespace: plugin.Namespace,
    Subsystem: "firewall",
    Name:      "blocked_total",
    Help:      "Counter of the number of requests blocked",
}, []string{"server", "zone"})

AllowedRequestsCount = prometheus.NewCounterVec(prometheus.CounterOpts{
    Namespace: plugin.Namespace,
    Subsystem: "firewall",
    Name:      "allowed_total",
    Help:      "Counter of the number of requests allowed",
}, []string{"server", "zone"})
```

The plugin will be documented and unit tested with the aim of 100% code coverage so that it is ready to be merged into the core CoreDNS repo

Timeline

Pre-Community Bonding Period:

Make as many contributions to the CoreDNS organization as possible. Read up more on DNS server.

Community Bonding Period (April 23rd - May 14th):

Keep making contributions to the core repo. Try writing a few CoreDNS plugins of my own to get more comfortable with the project

Coding Phase (May 27th - Aug 26th):

Coding Phase 1(May 27th - June 24th):

Week #	Tasks	Deliverables
Week 1 (May 27th - June2nd)	● Write a basic plugin to read plugin the firewall config from a Corefile	● The basic plugin code is merged
Week 2& 3 (June 3rd- June 16th)	● Implement methods for a basic block/allow mechanism that block individual IP strings defined in the Corefile ● Write phase1 blog post	● The block/allow mechanism is merged along with unit tests
Week 4 (June 17th - June 23rd)	● Buffer week ● Write docs if required	● Catch-up on any work left ● Solve a few issues in the core repo for fun

Coding Phase 2(June 25th - July 22nd):

Week #	Tasks	Deliverables
Week 1 & 2 (June 29th - July 12th)	● Enhance the plugin to read CIDR blocks from the Corefile instead of plain IP strings ● Change the block/allow methods to work on CIDR blocks	● The plugin recoginizes and works with CIDR blocks
Week 3 (June 13th - July 19th)	● Enable the plugin to read from blacklist/whitelist files ● Write the tests and docs	● The enhancements to the plugin are documented and

	● Write phase 2 blog post	merged along with unit tests
Week 4 (July 19nd - July 22nd)	● Buffer week. Catch-up on any work that is left	●

Coding Phase 3(July 9th - August 6th):

Week #	Tasks	Deliverables
Week 1 & 2 (July 27th - Aug 9th)	● Bump up the code coverage to 100% ● Implement metrics for the firewall plugin	● The test coverage stands at 100% ● Metrics feature is merged
Week 3 (Aug 10th - Aug 16th)	● Write the phase3 blog post ● Try to work on any stretch goals if possible	● Blog post is published
Week 4 (July 17th - Aug 26th)	● Buffer week ● Prepare the final project report	● Project is completed with the report submitted

Possible Outcomes:

- An external firewall plugin that is fully functional and unit tested
- The external plugin is ready to be merged into the coredns codebase as an internal plugin

Stretch goals / Future plans:

- I have been reading up on DNS amplification and flooding attacks and a few mitigation techniques have caught my eye for e.g. iptable rules, DNS response rate limiting. I am not sure how they will fit into the plugin or whether it will be possible or not but I would like to try this out in my own repo and maintain it as an external plugin if not merged into the core repo.
- I would like to keep contributing to CoreDNS as I am interested in making a career in the DevOps/Distributed Systems space and I think contributing to a CNCF project would help me gain the necessary insight and skills for this.

- Currently I am only trying to make PRs but I would also like to extend my contributions to raising issues and reviewing other PRs as I was doing with my previous GSoC community.

Why are you the right person to work on this project?

- I have ***past software development experience*** and have been working as a backend developer intern at http://appbase.io/ ***writing Go*** for about 6 months now so I am pretty comfortable with Go
- Last year ***I had completed a GSoC with coala.*** Here is my project completion report. So I am ***used to meeting the evaluation deadlines*** and ***submitting quality PRs***
- Have already started exploring the coreDNS project and I have ***made three PR*** to the core repo under review and been also trying my hand in ***writing a plugin*** by hacking on the demo plugin by Yong Tang. [Link to the plugin.]
- Have ***been contributing to open source software for over a year*** now so I am ***familiar with the git-flow*** and the best practices followed by various orgs.
- Have been researching and documenting my findings about my project and DNS servers in general since the last month here.
- The most important reason would be my love for open source software and communities and the desire to become a long term contributor to the communities that I contribute to.

Open Source Contributions

I am a regular open source contributor and have contributed to these organizations/projects (PR links are included) coala, Google, Ethereum Foundation, Kubernetes, Prometheus, Appbaseio, TaskCluster(Mozilla), DuckDuckGo, Kinto(Mozilla), Elastic, Kong, CoreDNS, Snowplow Analytics, NodeJS, OpenEBS and gojektech

Links:

- Github profile

- Linkedin profile
- Resume
- Openhub profile

Other Commitments

- **Do you have any other commitments during the GSoC period, May 8th to August 29th?**
  I will be interning at a startup from May13th to Aug13th and I would like to do GSoC along with my internship.
  I know that I will be able to handle the two together because last year I had almost 30-40 hrs of free time even after working on my project. I have successfully completed a GSoC so I understand the pressure of meeting deadlines and can handle it well. I will be working for 3-4 hours on my project during the weekdays and for the whole day (2*10 hrs) during the weekends.

- **Do you have exams or classes that overlap with this period?**
  No exams or classes during the coding phase.

- **Have you applied to any other organizations?**
  No I am only applying for the three coredns projects and no other organization. **The firewall plugin is my primary choice of project** I have submitted proposals for the azure DNS and google cloud DNS backend support as my 2nd and 3rd choices.