Golang to the rescue: Saving DevOps from TLS turmoil

GopherCon 2017 Lightning Talk

Chris Short
Manager of DevOps at Bankrate

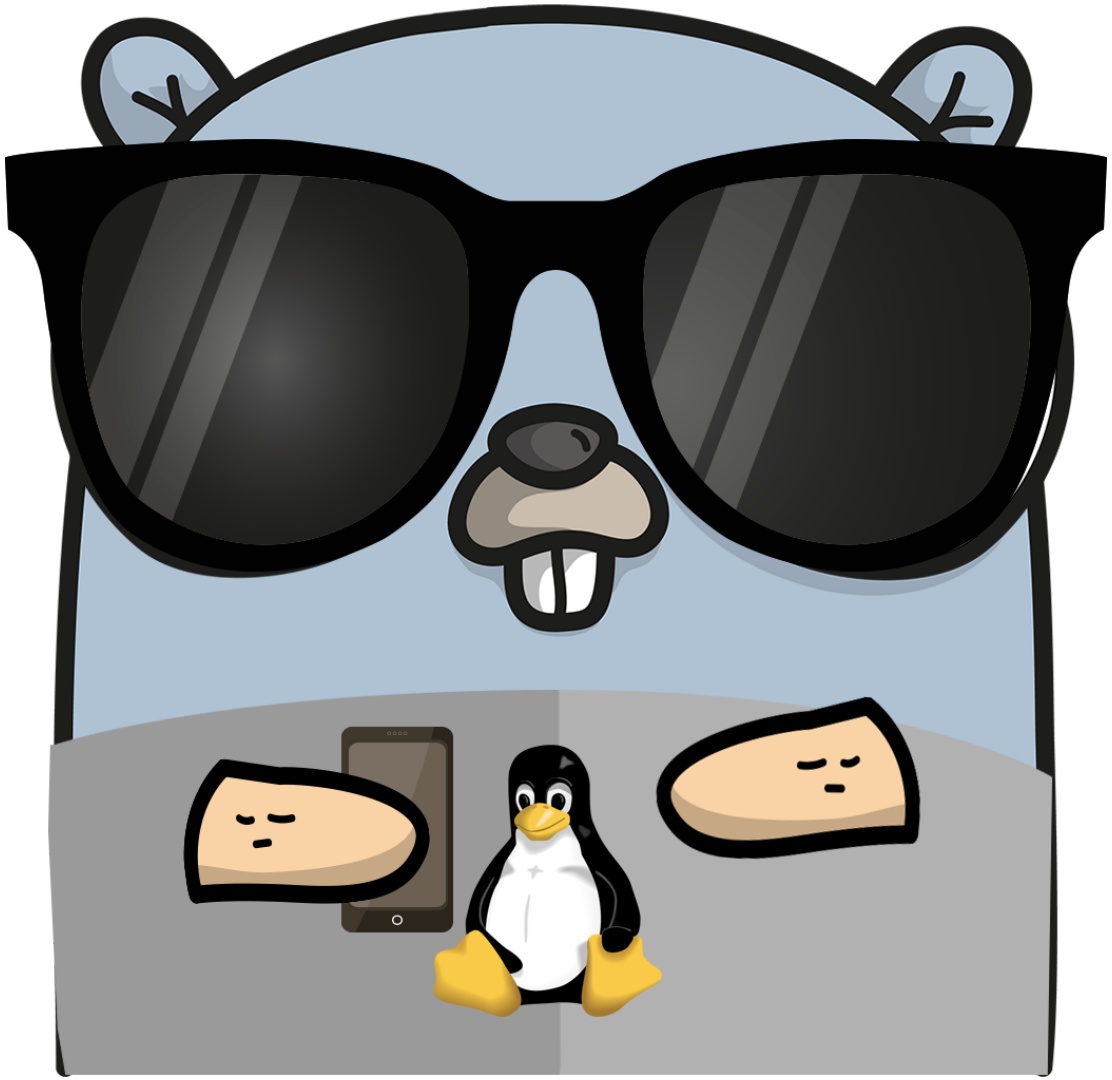
Introduction

Chris Short

- Manager of DevOps at Bankrate (http://www.bankrate.com)
- opensource.com (https://opensource.com/users/chrisshort) and DZone (https://dzone.com/users/2868764/chrisshort.html) Contributor
- Contributed to The Open Organization Guide to IT Culture Change (https://opensource.com/open-organization/resources/culture-change)
- DevOpsDays (https://www.devopsdays.org/) Speaker and Organizer
- DevOps'ish (https://devopsish.com/)
- chrisshort.net (https://chrisshort.net)
- @ChrisShort (https://twitter.com/ChrisShort)

This talk was derived from an opensource.com article I wrote in April 2017:
Golang to the rescue: Saving DevOps from TLS turmoil (https://opensource.com/article/17/4/testing-certificate-chains-34-line-go-program)

But Most Importantly



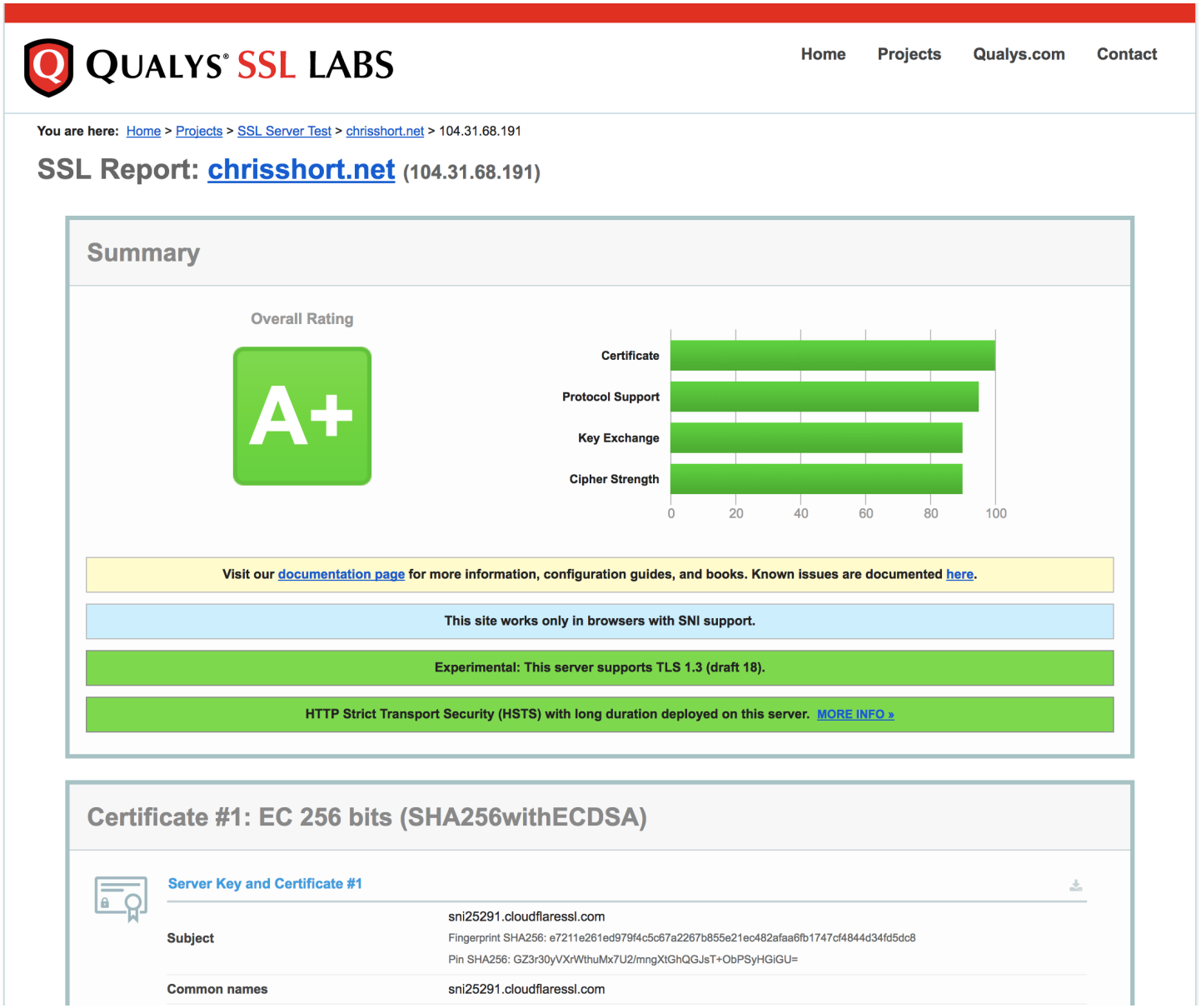
Me in Gopher Form by Gopherize.me (https://gopherize.me/)

Not Too Long Ago in a Place of Work Far, Far Away...

Let's Talk Certificate Chains

2 Chainz (we can talk rap music later)

This is the Goal



NBD … OMG

RapidSSL.
Simple site security for less.

Regions: US/Canada Europe UK Australia

buy | switch | resell | learn | support

Home > Support > General Information Details

RapidSSL Intermediate CAs

Printable version

General Information ID: INFO1548 Updated: 06/02/2016

Description

RapidSSL uses an Intermediate CA to enhance the security of SSL certificates. When installing a RapidSSL certificate, it is essential to install the Intermediate CA at the same time as the SSL certificate, this ensures that the SSL certificate is fully trusted by all browsers and prevents SSL errors from appearing when users visit the website.

The RapidSSL and Wildcard certificates Intermediate CA can be downloaded from the table below. Once your SSL certificate is installed, please use the SSL Certificate Installation Checker to ensure a problem-free SSL certificate experience for you and your users!

CONTACT SUPPORT

US Support:

Order Processing
Email Form

Technical Support
Email Form

European Support:

Order Processing

RSA SHA-1 SSL Certificates

Product Name	Intermediate CA
RapidSSL Wildcard FreeSSL	SO26462

RSA SHA-2 (under SHA-1 Root) SSL Certificates

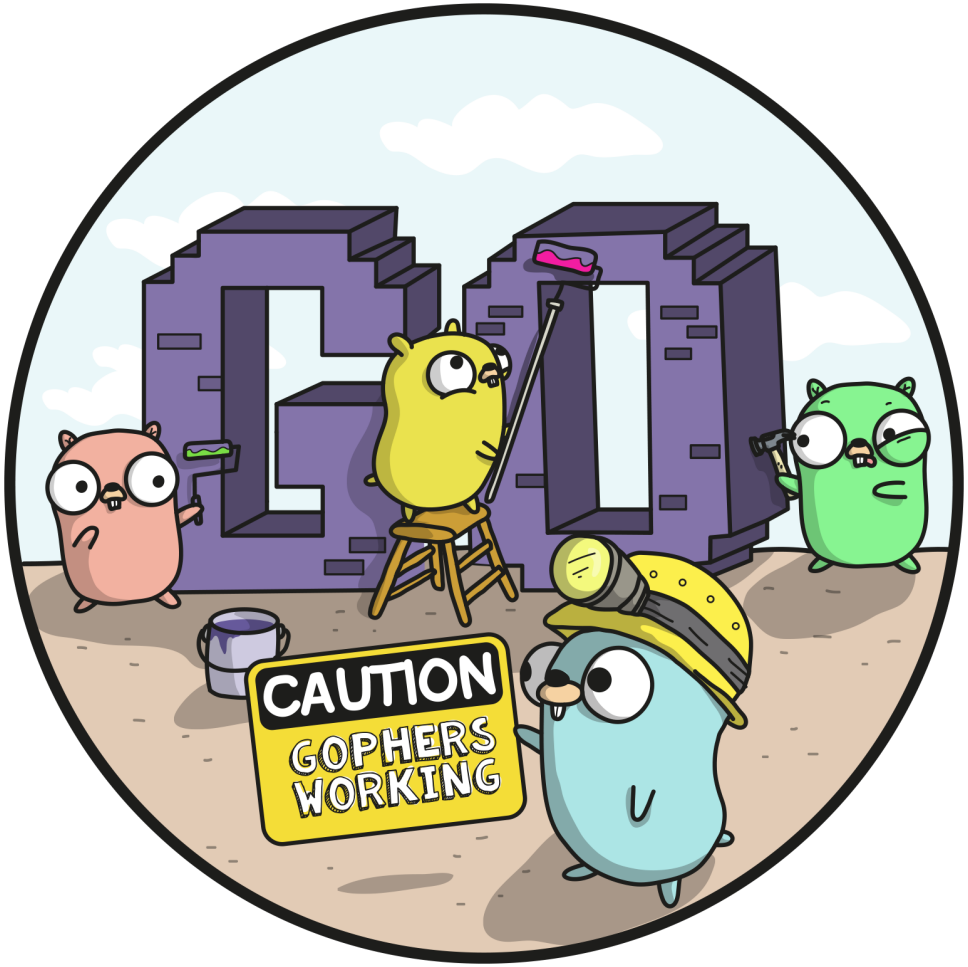
Product Name	Intermediate CA
RapidSSL Wildcard FreeSSL	SO28616

RSA SHA-2 (under SHA-2 Root) SSL Certificates

digicert® SSL Solutions Partner Company Support 1.801.701.9600

Compatibility Intermediate CA	Issuer: GTE CyberTrust Global Root Valid until: 10/Aug/2018 Serial #: 0E:E0:68:2D:BB:98:2D:92:C6:85:6A:DA:DE:48:19:80 Thumbprint: F08B49D0EBE7975062CD19C731B141DF4D11DF52 Download
Cybertrust Japan Issuing CA-1	Issuer: Verizon Global Root CA Valid until: 01/Sep/2026 Serial #: 0C:5B:12:0D:AC:42:A1:CB:7B:20:89:DB:17:6E:04:78 Thumbprint: 4B8FE3B160D85B627F660C6A425059C2A420A774 Download
DigiCert Assured ID CA-1	Issuer: DigiCert Assured ID Root CA Valid until: 10/Nov/2021 Serial #: 06:FD:F9:03:96:03:AD:EA:00:0A:EB:3F:27:BB:BA:1B Thumbprint: 19A09B5A36F4DD99727DF783C17A51231A56C117 Download
DigiCert Assured ID CA G2	Issuer: DigiCert Assured ID Root G2 Valid until: 01/Aug/2028 Serial #: 0F:5F:CC:FC:AB:20:F3:DF:8E:6D:A3:D8:47:67:C2:93 Thumbprint: 28E96CDB1DBA273FD1A6151BE15F088F26046273 Download
DigiCert Assured ID CA G3	Issuer: DigiCert Assured ID Root G3 Valid until: 01/Aug/2028 Serial #: 01:05:DA:E2:55:AA:B2:95:4A:0D:B2:C9:E6:B5:32:2C Thumbprint: C619BE4F415453F46D020ED79F5D5CA5C37E14AD Download
DigiCert Assured ID Code Signing CA-1	Issuer: DigiCert Assured ID Root CA Valid until: 10/Feb/2026 Serial #: 0F:A8:49:06:15:D7:00:A0:BE:21:76:FD:C5:EC:6D:BD Thumbprint: 409AA4A74A0CDA7C0FEE6BD0BB8823D16B5F1875 Download

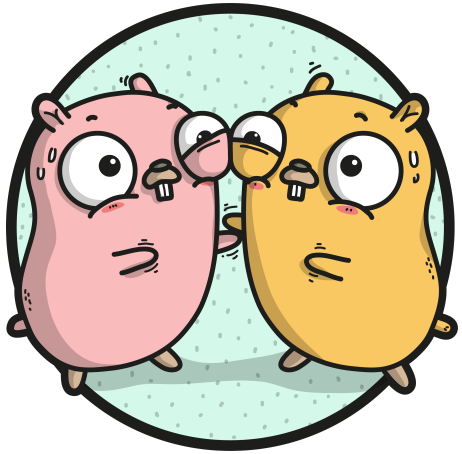
So What Does Any Good Engineer Do?



Go Build by Ashley McNamara (https://github.com/ashleymcnamara/gophers)

Three Go Packages: log

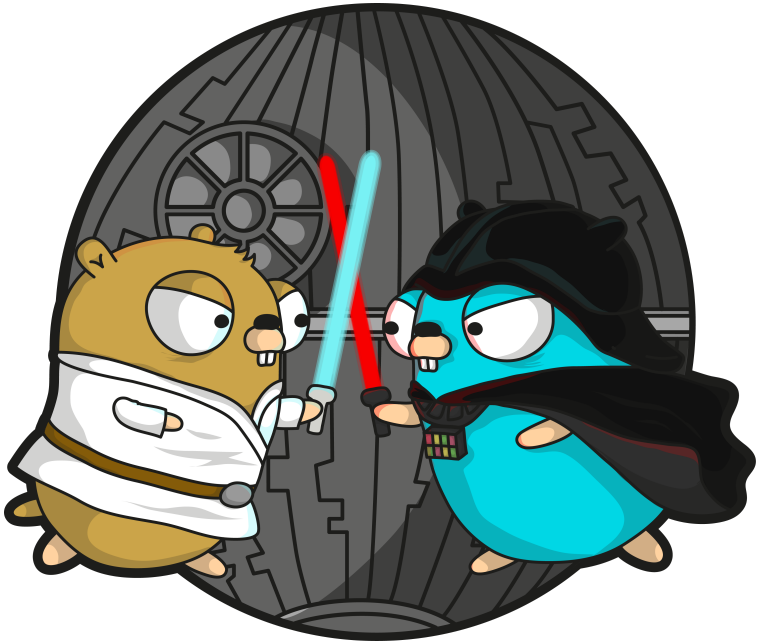
- The go `log` (https://golang.org/pkg/log/) package is pretty self explanatory
- Package that enables logging
- Needed a spectacular failure at the sign of trouble
- `log` has three helper functions: `print`, `fatal`, and `panic`
- Output from the package goes to `stderr`
- Used a `fatal` error to get the web server to stop and log any issue



Hugging Gophers by Ashley McNamara (https://github.com/ashleymcnamara/gophers)

Three Go Packages: crypto/tls

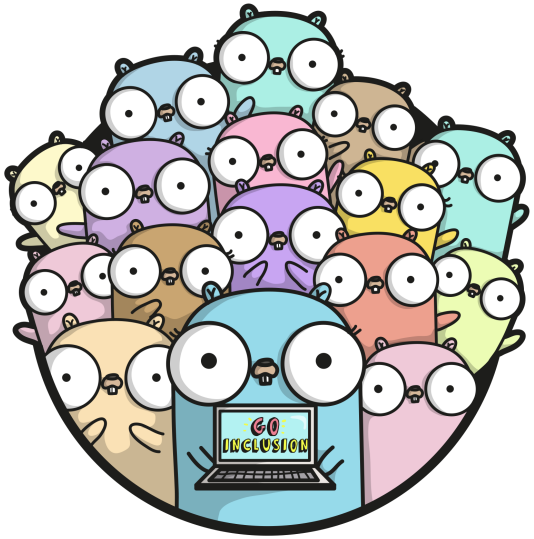
- The Go `crypto/tls` (https://golang.org/pkg/crypto/tls/) package partially implements TLS 1.2, as specified in RFC 5246 (https://tools.ietf.org/html/rfc5246)
- Package configures usable SSL/TLS versions
- Identifies preferred cipher suites and elliptic curves used during handshakes
- This is the package that handles connections securely



Gopher Star Wars by Ashley McNamara (https://github.com/ashleymcnamara/gophers)

Three Go Packages: net/http

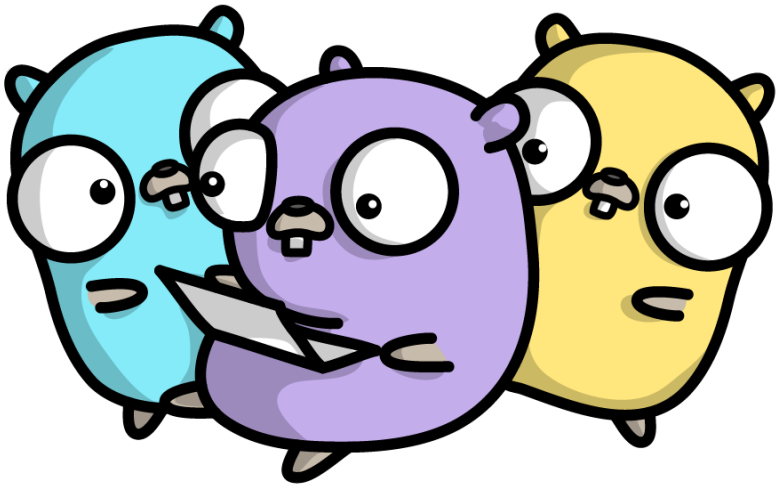
- Go implementation of HTTP
- `net/http` (https://golang.org/pkg/net/http/) has a function called `ListenAndServeTLS`
- ListenAndServeTLS provides the desired certificate checking functionality
- "If the certificate is signed by a certificate authority, the certFile should be the concatenation of the server's certificate, any intermediates, and the CA's certificate."



Gopher Inclusion by Ashley McNamara (https://github.com/ashleymcnamara/gophers)

main: mux, cfg, srv

- Code creates a `mux`, short for HTTP request multiplexer
- I multiplexers (it's a long story that involves analog signals)
- `mux` has a function that creates an HTTP server with headers and content (Hello World!)
- `cfg` brings in all the TLS bits seen in a solid web server config
- `srv` puts the pieces together and defines what port to listen on



Gopher Share by Ashley McNamara (https://github.com/ashleymcnamara/gophers)

Fail Spectacularly

- I DevOps
- I embrace failure
- `log.Fatal(srv.ListenAndServeTLS("/etc/ssl-tester/tls.crt", "/etc/ssl-tester/tls.key"))`
- Defines path of certificate files to use
- Also logs a fatal error if certificate is not valid
- Fails Fast

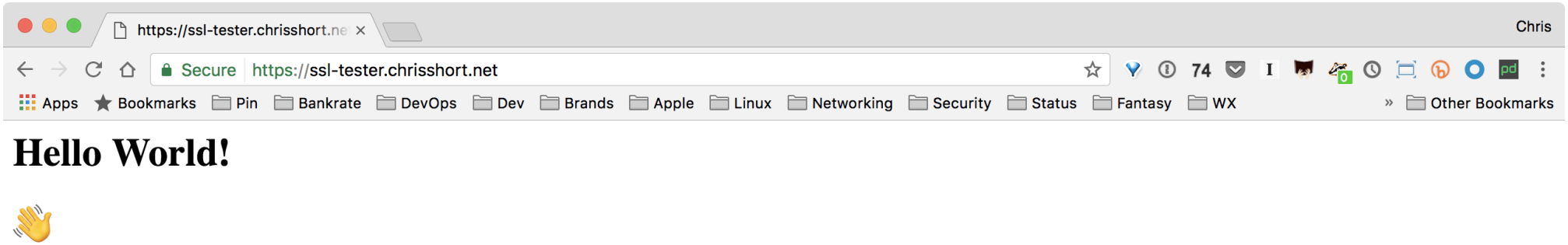
It's Open Source!

41 lines (38 sloc) | 1.28 KB — Sourcegraph | Raw | Blame | History

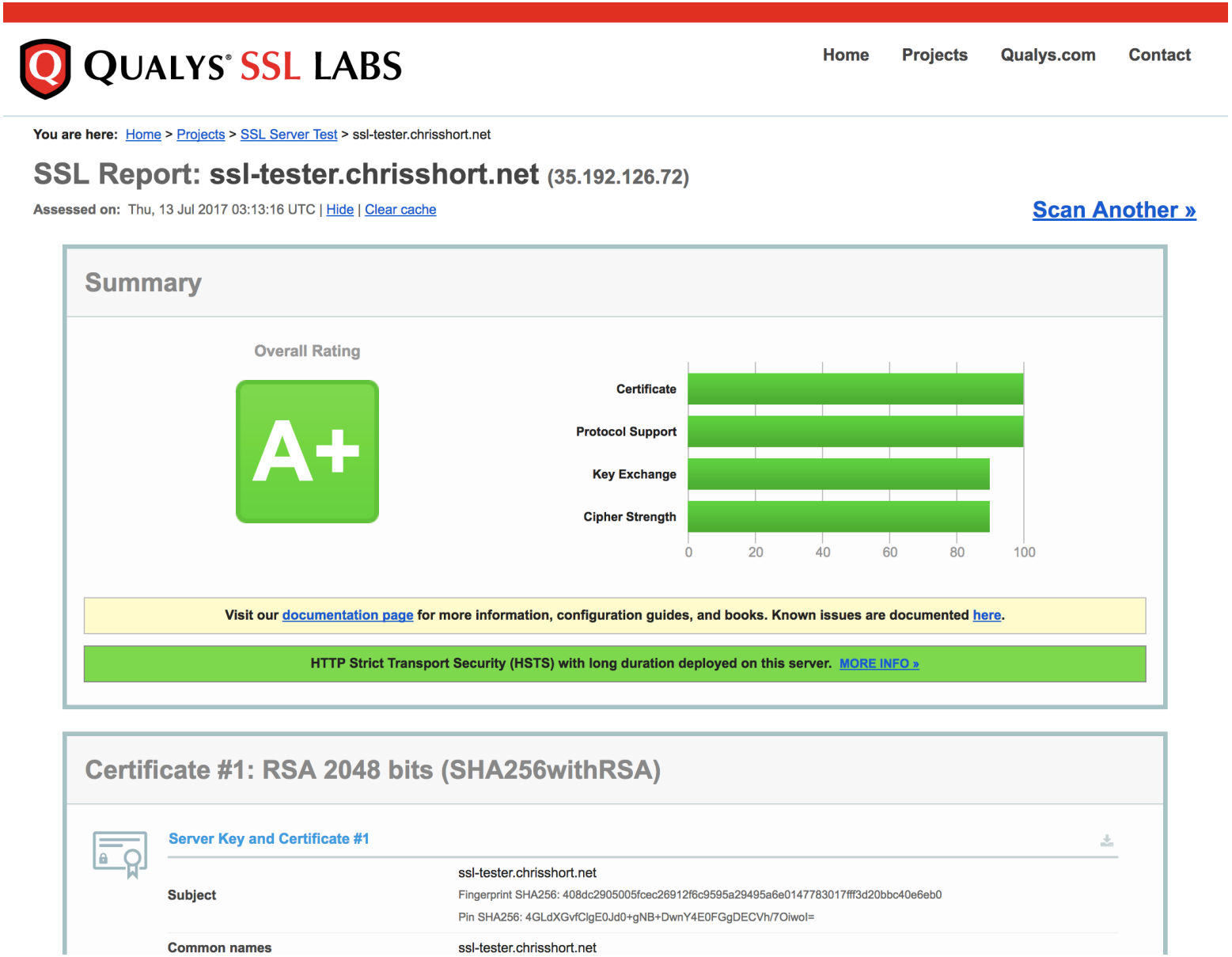
```go
package main

import (
	"crypto/tls"
	"log"
	"net/http"
)

func main() {
	mux := http.NewServeMux()
	mux.HandleFunc("/", func(w http.ResponseWriter, req *http.Request) {
		w.Header().Add("Strict-Transport-Security", "max-age=63072000;")
		w.Write([]byte("<h1>Hello World!</h1>\n<h1>👋</h1>"))
	})
	cfg := &tls.Config{
		MinVersion:               tls.VersionTLS12,
		CurvePreferences:         []tls.CurveID{tls.CurveP521, tls.CurveP384, tls.CurveP256},
		PreferServerCipherSuites: true,
		CipherSuites: []uint16{
			tls.TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
			tls.TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
			tls.TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
			tls.TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
			// POLY1305 ciphers are not in Go 1.6 and 1.7
			//                     tls.TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305,
			//                     tls.TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305,
			tls.TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
			tls.TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
			tls.TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
			tls.TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
		},
	}
	srv := &http.Server{
		Addr:         ":443",
		Handler:      mux,
		TLSConfig:    cfg,
		TLSNextProto: make(map[string]func(*http.Server, *tls.Conn, http.Handler), 0),
	}
	log.Fatal(srv.ListenAndServeTLS("/etc/ssl-tester/tls.crt", "/etc/ssl-tester/tls.key"))
}
```

https://github.com/chris-short/ssl-tester (https://github.com/chris-short/ssl-tester)

It Works!

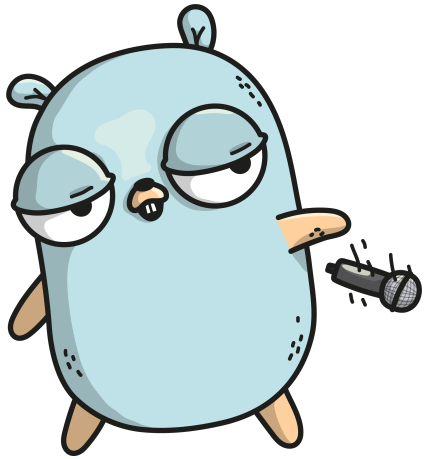

https://ssl-tester.chrisshort.ne
Chris
Secure
https://ssl-tester.chrisshort.net
74
Apps
Bookmarks
Pin
Bankrate
DevOps
Dev
Brands
Apple
Linux
Networking
Security
Status
Fantasy
WX
Other Bookmarks
Hello World!
👋

No. It Really Works!



Conclusion

- The Go code does exactly what I need it to do and nothing more
- About 40 lines of code!!! I Go!
- Binary is a self contained web server
- Less than 6MB!!! I Go!
- Can be safely deployed to any public server
- External testing run against it for extra vetting



Gopher Mic Drop by Ashley McNamara (https://github.com/ashleymcnamara/gophers)

Thank you

Chris Short
Manager of DevOps at Bankrate
chris@chrisshort.net (mailto:chris@chrisshort.net)
https://devopsish.com (https://devopsish.com)
@ChrisShort (http://twitter.com/ChrisShort)