Audit Checklist

Go – Version 1.1

Send us the scope for the audit:

- [ ] Send us relevant documentation such as whitepapers, websites, or online documentation
- [ ] Send us the repositories and commit hash that should be used for the quote
- [ ] Specify the relevant directories/files in the repositories for the audit
- [ ] Send us an outline of how the scope/code will change before the audit begins, including a detailed description of any still unimplemented parts

In addition, before the audit starts:

- [ ] Resolve any compiler warnings (e.g. `make` or `go build`)
- [ ] Apply code formatting (e.g. `make format` or Gofmt `gofmt -w .`)
- [ ] Resolve any issues returned by your linter (e.g. `make lint` or golangci-lint: `golangci-lint run`)
- [ ] Resolve any issues returned by Vet: `go vet`
- [ ] Ensure all tests pass (e.g. `make test` or `go test`)
- [ ] Ensure you have high test coverage. To start an audit, we require at least 70%, ideally 90% test coverage. You can compute test coverage, e.g., through go tool cover: `go test -cover`
- [ ] Remove any files that are not specific to your project, e. g. Markdown files from a template you used
- [ ] Remove any commented/unused code
- [ ] Resolve any pending TODOs
- [ ] Send us the repositories and commit hash that should be used for the audit (code freeze)
- [ ] Specify the relevant directories/files in the repositories for the audit
- [ ] Send us the computed test coverage

Oak Security
https://oaksecurity.io/
info@oaksecurity.io