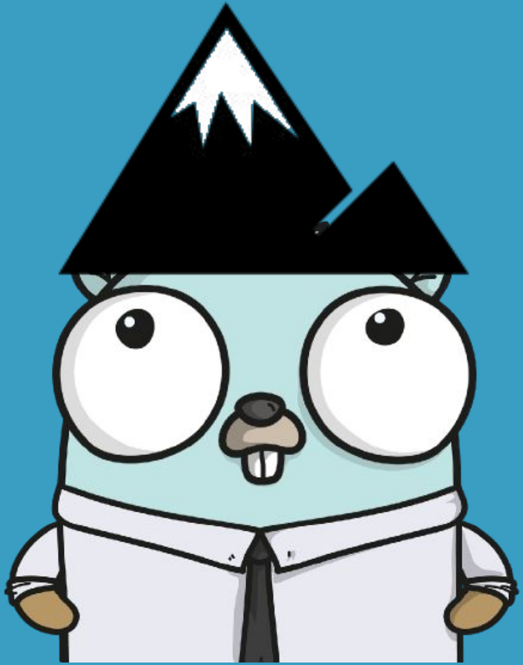

Go-ing Secure
at ironPeak

testament to Go

fosdem 2021

1. $ tree

- whoami
- code
- app
- system
- risk
- tl;dr

2. $ whoami

Niels Hofmans

role	cybersecurity freelancer
work	code, app, system security & risk
likes	go, open source, cloud native, media
contact	hello@ironpeak.be
website	ironpeak.be
github	github.com/hazcod

$ game rules

4X

Problem

Solution

Good & Bad

$ code

3. $ code

Problem

developers write insecure code

3. $ code

Problem

developers write insecure code

Solution

go is opinionated & puristic

go/parser, golangci-lint, codeql (semml), ...

3. $ code

Problem

developers write insecure code

Solution

go is opinionated & puristic

go/parser, golangci-lint, codeql (semml), ...

Good

go **is** opinionated

Bad

go is **opinionated**

3. $ code

Problem

supply chain attacks

3. $ code

Problem

supply chain attacks

Solution

go embraces open source

github.com/golang, modules

3. $ code

Problem

supply chain attacks

Solution

go embraces open source

github.com/golang, modules, inline docs

Good

open source

Bad

open **source**

3. $ code

Problem

it’s either slow garbage collection or memory vulns

3. $ code

Problem

it’s either slow garbage collection or memory vulns

Solution

go runs an efficient garbage collector

opentelemetry, pprof and delve are awesome

3. $ code

Problem

it's either slow garbage collection or memory vulns

Solution

go runs an efficient garbage collector

opentelemetry, pprof and delve are awesome

Good

no need to worry

Bad

no **need** to worry

3. $ code

Problem

open source dependency hell

3. $ code

Problem

open source dependency hell

Solution

go module proxy, forking, commit pinning

go.sum, GOPROXY, GOPRIVATE

3. $ code

Problem

open source dependency hell

Solution

go module proxy, forking, commit pinning

go.sum, GOPROXY, GOPRIVATE

Good

git-focused public proxy

Bad

git-focused **public** proxy

```
$ app
```

4. $ app

Problem

what is a secure application?

4. $ app

Problem

what is a secure application?

Solution

OWASP ASVS, OWASP Go-SCP, awesome-golang-security, paseto, ...

4. $ app

Problem

what is a secure application?

Solution

OWASP ASVS, OWASP Go-SCP, awesome-golang-security, paseto, ...

Good

loads of resources available

Bad

loads of resources available

4. $ app

Problem

hands-down nature of Go results in insecure applications

4. $ app

Problem

hands-down nature of Go results in insecure applications

Solution

provide secure defaults, awareness, SAST

google/go-safeweb (WIP)

4. $ app

Problem

hands-down nature of Go results in insecure applications

Solution

provide secure defaults, awareness, SAST

google/go-safeweb (WIP)

Good

no-fuzz Go technicals

Bad

no-fuzz Go **technicals**

4. $ app

Problem

secure application development is slow

4. $ app

Problem

secure application development is slow

Solution

Go skeleton templates with module wrappers

4. $ app

Problem

secure application development is slow

Solution

Go skeleton templates with module wrappers

Good

rapid prototyping

Bad

Go **malware** payloads

4. $ app

Problem

secure application development is slow

Solution

Go skeleton templates with module wrappers

Good

rapid prototyping

Bad

Go **malware** payloads

4. $ app

Problem

WAFs are contextless

4. $ app

Problem

WAFs are contextless

Solution

Runtime Application Self-Protection (RASP) is on the rise

sqreen, ...

4. $ app

Problem

WAFs are contextless

Solution

Runtime Application Self-Protection (RASP) is on the rise

sqreen, ...

Good

runtime threat **detection**

Bad

runtime threat detection

$ system

4. $ system

Problem

microservices are a PITA to maintain

4. $ system

Problem

microservices are a PITA to maintain

Solution

service meshes such as linkerd, istio + telemetry

temporal is awesome

4. $ system

Problem

microservices are a PITA to maintain

Solution

service meshes such as linkerd, istio + telemetry

temporal is awesome

Good

observability is key

Bad

observability is **key**

4. $ system

Problem

huge network attack surface

4. $ system

Problem

huge network attack surface

Solution

zero trust topology

wireguard-go, tailscale, cloudflare teams

4. $ system

Problem

huge network attack surface

Solution

zero trust topology

wireguard-go, tailscale, cloudflare teams

Good

one **method** of entry

Bad

one method of entry

4. $ system

Problem

doing secrets right is hard

4. $ system

Problem

doing secrets right is hard

Solution

vault, sealed-secrets, secrets-manager, …

4. $ system

Problem

doing secrets right is hard

Solution

vault, sealed-secrets, secrets-manager, ...

hashicorp/vault, enpass-cli, ...

Good

one vault everywhere

Bad

one vault **everywhere**

4. $ system

Problem

smart home peripherals are insecure

4. $ system

Problem

smart home peripherals are insecure

Solution

linux kernel + Go + bruttela/hc (HomeKit)

gokrazy, tinygo, tamago

4. $ system

Problem

smart home peripherals are insecure

Solution

linux kernel + Go + bruttela/hc (HomeKit)

gokrazy, tinygo, tamago

Good

just **Go**

Bad

just Go

$ risk

5. $ risk

Problem

on-device attack surface

5. $ risk

Problem

on-device attack surface

Solution

static go builds, read-only, empty container images, hardened k8s runtime, hardened Pods

5. $ risk

Problem

on-device attack surface

Solution

static go builds, read-only, empty container images, hardened k8s runtime, hardened Pods

Good

difficult exploitation

Bad

go/glibc version, tzdata, ca certs

5. $ risk

Problem

servicing kubernetes is a wildfire

5. $ risk

Problem

servicing kubernetes is a wildfire

Solution

open-policy-agent policies with gatekeeper

5. $ risk

Problem

servicing kubernetes is a wildfire

Solution

open-policy-agent policies with gatekeeper

Good

anything **can** run on k8s

Bad

anything can run on k8s

5. $ risk

Problem

pentesting is complex & hard

5. $ risk

Problem

pentesting is complex & hard

Solution

a **lot** of Go open-source security tools

5. $ risk

Problem

pentesting is complex & hard

Solution

a **lot** of Go open-source security tools

Good

open **source** tooling

Bad

open source tooling

5. $ risk

Problem

deployment needs to go fast nowadays

5. $ risk

Problem

deployment needs to go fast nowadays

Solution

embed cybersecurity fast & left

5. $ risk

Problem

deployment needs to go fast nowadays

Solution

embed cybersecurity fast & left

Good

devs become **SREs**

Bad

devs become SREs

TL;DR

6. tl;dr

Challenges

- cybersecurity is changing **fast**
- **complex** infrastructure abstraction
- **cloud** native

6. tl;dr

Challenges

- cybersecurity is changing **fast**
- **complex** infrastructure abstraction
- **cloud** native

Hope

- cybersecurity is **changing** fast
- complex infrastructure **abstraction**
- cloud **native**

6. tl;dr

thanks to Go