The Church of Golang (OWASP Fortaleza)

Primeira Missa
Dr. Victor Pasknel

Victor Pasknel

- Doutor em Informática Aplicada @ Unifor
- Pentester / Red Team (~13 anos)
- Pesquisador
- Professor Universitário
- Baterista
- Medium: @pasknel
- Palestrante:
  - H2HC / BSides SP / NullByte
  - BWCON / JampaSec / CAJUSec
  - Python Brasil / Python Nordeste
  - DevOps Days Fortaleza
  - Roadsec / MindTheSec (SP & RJ)

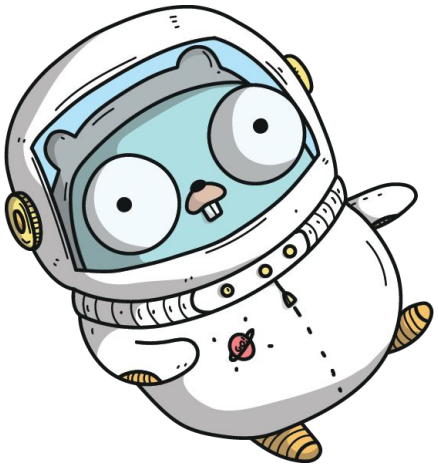
Agenda

- 0x01 - Introdução
- 0x02 - Instalação & Configuração
- 0x03 - Sermão do Dia

Introdução

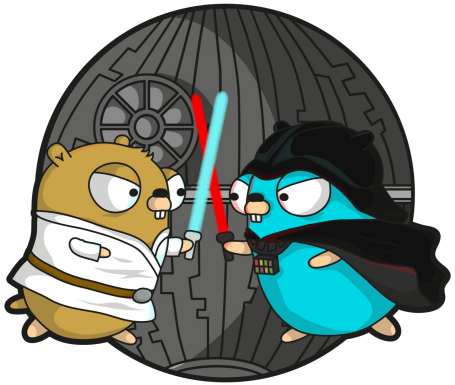
Introdução

- **Minha jornada com Golang**
  - **2017:** Previamente usando Python (~7 anos)
  - **2018:** Aprendendo Golang no doutorado
  - **2020:** Desenvolvimento de protótipo em Golang
  - **2022:** Utilizando Golang em Pentests



Introdução

- Linguagem de programação criada pelo Google
  - Lançamento do projeto: Novembro/2009
  - Primeira versão: Março/2012
- Características principais:
  - Código aberto (open source)
  - Linguagem compilada
  - Tipagem estática
  - Programação concorrente nativa
  - Simplista
    - Foco na **velocidade**!
    - Syntax limpa
    - Poucas palavras reservadas



Instalação & Configuração

Instalação & Configuração

- Linux (Ubuntu):

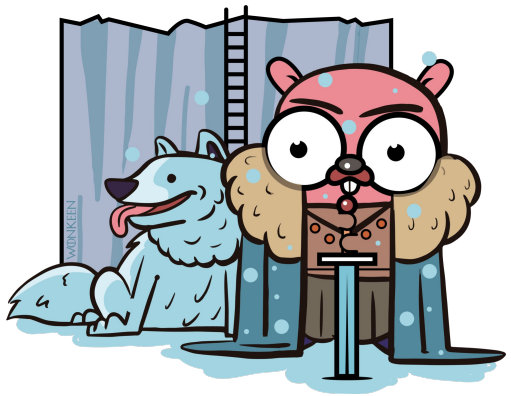
```
$ sudo apt-get update
$ sudo apt-get install golang-go
$ go version
```

- Mac:

```
$ brew install go
$ go version
```

- Docker:

```
$ docker container run -it golang:1.18
```



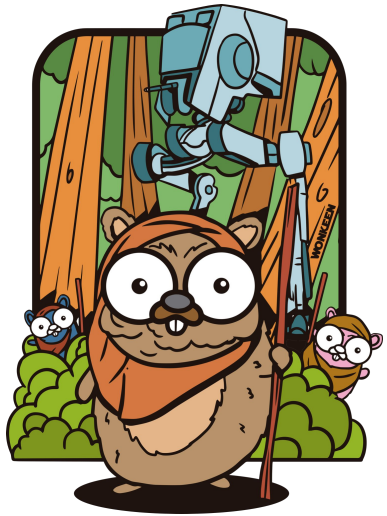
Instalação & Configuração

- Variáveis de Ambiente:
  - **$GOROOT:** Path do compilador (e outras tools)
  - **$GOPATH:** Path do Go Workspace
- Verificando variáveis:

```
$ go env
```

- Exportando **GOPATH** (*caso necessário*):

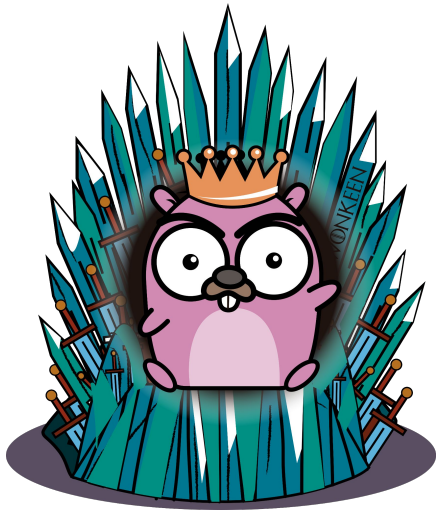
```
$ export GOPATH=$HOME/go
$ export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin
```



Instalação & Configuração

- Go Workspace (GOPATH)
  - **GOPATH** contém 3 diretórios:
    - **bin:** contém binários (que o *go install* compila)
    - **pkg:** contém arquivos de objeto pré-compilados
    - **src:** contém todos os nossos códigos-fontes
  - Estrutura de diretórios:

```
.
├── bin
├── pkg
└── src
    └── github.com/foo/bar
        └── bar.go
```



Sermão do Dia

Sermão do Dia

- User Enumeration (OneDrive)
  - Técnica de enumeração de usuários via OneDrive (e Sharepoint)
  - Vamos criar um script de prova de conceito (PoC)
  - Fonte Original:
    - https://www.trustedsec.com/blog/onedrive-to-enum-them-all/

Sermão do Dia: User Enumeration (OneDrive)

- Endpoint (Sharepoint)
  - Para usuário do OneDrive, uma pasta é criada no Sharepoint:

```
https://<tenant>-my.sharepoint.com/personal/<UserPrincipalName>/_layouts/15/onedrive.aspx
```

  - Informações necessárias:
    - **Tenant**: Costuma ser o nome da empresa
    - **UserPrincipalName**: Email do funcionário
      - **Trocar '.' e '@' por '_'**

Sermão do Dia: User Enumeration (OneDrive)

- Prova de Conceito (PoC):
  - Pré-requisito:
    - Lista de nomes de usuários
  - Para cada usuário:
    - Criar uma requisição para o endpoint do Sharepoint
    - Criar um client HTTP e enviar requisição
    - Verificar **StatusCode** da resposta
    - Apresentar o resultado

Sermão do Dia: User Enumeration (OneDrive)

- Prova de Conceito (PoC):
  - Arquivo: */PoC01/main.go*
  - Objetivos da PoC:
    - Hello World básico
    - Estrutura básica de um programa
    - Função de escrita (*fmt.Println*)
    - Como criar um módulo
    - Como compilar o projeto

Sermão do Dia: User Enumeration (OneDrive)

- Comandos Básicos:
  - Criando um novo módulo:

```
$ go mod init github.com/SEU_USUARIO/SEU_PROJETO
```

  - Download de dependências:

```
$ go mod tidy
```

  - Compilar projeto (build):

```
$ go build
```

Sermão do Dia: User Enumeration (OneDrive)

- Cross-Compile
  - Variáveis de ambiente
    - **GOARCH**: Arquitetura
    - **GOOS**: Sistema Operacional
  - Lista de GOOS/GOARCH

```
$ go tool dist list
```

  - Exemplos:

```
$ GOOS=windows GOARCH=amd64 go build
$ GOOS=darwin GOARCH=amd64 go build
```

Sermão do Dia: User Enumeration (OneDrive)

- Prova de Conceito (PoC):
  - Arquivo: */PoC02/main.go*
  - Objetivos da PoC:
    - Importar bibliotecas nativas (ex: *net/http*)
    - Criar um client HTTP
    - Criar um requisição HTTP
    - Enviar requisição
    - Verificar resposta
    - Tratamento de erros

Sermão do Dia: User Enumeration (OneDrive)

- Prova de Conceito (PoC):
  - Arquivo: */PoC03/main.go*
  - Objetivos da PoC:
    - Entender como funções são criadas em Golang
    - Passagem de parâmetros e valores de retorno
    - Criando uma função de enumeração

Sermão do Dia: User Enumeration (OneDrive)

- Prova de Conceito (PoC):
  - Arquivo: */PoC04/main.go*
  - Objetivos da PoC:
    - Leitura de arquivos
    - Interpolação de strings

Dúvidas ?

pasknel@gmail.com