CiscnBuild 设计报告

### 目录结构

├── CiscnMap.go 运行入口  
├── README.md  
├── config  
│   └── Config.go 配置文件,一些全局配置在此  
├── go.mod  
├── go.sum  
├── ip.txt 存放IP地址,如165.22.92.0/24  
├── result.json 运行的结果  
└── utils  
├── Device.go 设备识别  
├── HoneyPot.go 蜜罐识别  
├── HostAlive.go 主机存活检测  
├── IPParese.go 解析ip.txt,放入主机存活检测中  
├── Output\_Json.go 打开文件,并写入文件格式  
├── PortScan.go 端口扫描  
├── Protocol.go 协议识别  
├── ServiceDectet.go 指纹识别  
└── func.go 存储一些常用函数

### 设计思路

1. 解析ip.txt,将ip放入到主机存活队列中
2. 对于ip,进行多线程主机存活检测
3. 如果存活,对端口进行多线程扫描,设备识别,蜜罐识别
4. 如果该端口打开,对该端口进行协议识别与指纹识别
5. 所有信息写入ipInfo这个map类型,然后写入文件result.json

### 具体实现

#### 设备识别

针对每个IP,对设备的特定端口进行扫描,如果打开,根据服务器回应进行端口识别

#### 蜜罐识别

针对每个IP,对蜜罐的默认端口进行扫描,针对在特定的回显和上下文的蜜罐特征,对蜜罐进行识别

#### 协议识别

先根据默认端口给出协议,然后再根据回应报文进一步判断

#### 服务识别

根据server字段,ssh的回应,等等不同协议的不同特征进行判断

#### 主机存活

根据ICMP协议对主机存活性进行判断

#### 端口扫描

采用tcp协议根据回应来判断端口状态