بسم اللّه الرحمن الرحیم

**Project Description:**
**Network Scanner for Open Ports and Banner Grabbing**

## مسیر پروژه

آشنایی با مفاهیم و اصطلاحات متن پروژه

نصب نرم افزار Nmap و یادگیری چند کامند مورد نیاز

تماشای یک دوره کوتاه و آشنایی با سینتکس زبان Golang

سرچ و تحقیق درباره نحوه پیاده سازی و کتابخانه های مورد نیاز

## مفاهیم

- **پورت**

  مسیر هایی در شبکه برای ارتباط با سرویس های مختلف که به هریک از آن ها یک شماره از 0 تا 65535 اختصاص یافته است

  FTP(21) پورت مخصوص ارسال فایل

  SMTP(587,25) پورت مخصوص ارسال ایمیل

  پورت های با اهمیت دیگر شامل HTTP, HTTPS, SHH و... هستند

- بنر گربینگ **(Banner Grabbing)**

  برخی از پروت ها هنگام اتصال یک پیام ارسال می کنند که که شامل اطلاعاتی درباره آن پورت اصلا به طور مثال می توان شامل ورژن نرم افزار و قابلیت های ان باشد (فرضا برای STMTP ارسال ایمیل قابلیت مورد نظر است)

  ما در این پروژه سعی داریم این بنرها را از پورت های FTP, SMTP دریافت کنیم

- **TCP Connection**

  TCPیک نوع پرتکول ارتباطی است و در این پروژه از net.DialTimeout برای برقراری اتصال با پورت استفاده شده است

  Timeoutبه مدت زمانی اشاره دارد که برنامه تلاش می کند با سرور ارتباط برقرار کند و اگر در این مدت زمان پاسخی از سرور دریافت نشود اتصال قطع می شود

## توضیحات خط به خط کد پروژه

```go
package main

import (
    "fmt"
    "net"
    "strconv"
    "strings"
    "sync"
    "time"
)
```

ایمپورت کتابخانه هاب مورد نیاز که مهم ترین آن ها net که برای کار با شبکه و اتصال به پروت ها استفاده می شود و Time  که برای کار با زمان و به طور خاص برای این پروژه مشخص کردن حداکثر زمان انتظار برای برای دریافت پاسخ از سرور به کار گرفته شده است .

همچنین Sync نیز برای پیاده سازی قابلیت مولتی تردینگ به کارگیری شده است.

## توابع

- CreateIPRange: IP آغازین و پایانی را دریافت می کند و رنج ای پی هایی را که باید بررسی شوند به صورت یک آرایه برمیگرداند
- ScanPort : این تابع IP و پورت مورد نظر را دریافت می کند و مشخص می کند که پورت مورد نظر باز است یا خیر همچنین در صورت باز بود پورت های FTM, SMTP بنر آن ها را دریافت می کند
- BannerGrabbing: این تابع بنر را از پورت مربوطه دریافت می کند

```go
func main() {
    var startIP string
    var endIP string
    var timeoutInt int

    var ports []int = []int{21, 22, 25, 80, 443, 587}

    fmt.Print("Enter start IP: ")
    fmt.Scan(&startIP)
    fmt.Print("Enter end IP: ")
    fmt.Scan(&endIP)
    fmt.Print("Enter timeout (seconds): ")
    fmt.Scan(&timeoutInt)

    var timeout time.Duration = time.Duration(timeoutInt) * time.Second

    var IPRange []string = CreateIPRange(startIP, endIP)

    var waitGroup sync.WaitGroup
    for i := 0; i < len(IPRange); i++ {
        for j := 0; j < len(ports); j++ {
            waitGroup.Add(1)
            go ScanPort(IPRange[i], ports[j], timeout, &waitGroup)
        }
    }

    waitGroup.Wait()

    fmt.Println("End!")
}
```

در آرایه Ports چند پورت شماره چند پورت با اهمیت نگهداری میشود که برنامه آن ها را بررسی می کند در صورت نیاز می توان پورت هایدیگر را به این آرایه اضافه کرد

ابتدا ای پی آغازین و پایانی به همراه مدت زمان انتظار برای دریافت اطلاعات از سرور را از کاربر دریافت می کنیم

Timeoutدریافت شده از کاربر از جنس عددی است پس باید آن را با استفاده از time.Duration به زمان تبدیل کنیم

سپس با استفاده از تابع CreateIPRange بازه ای از ای پی هایی که باید مورد بررسی قرار گیرند را می سازیم و در یک آرایه ذخیره می کنیم

متغیر waitGroup را برای مدیریت Goroutine ها می سازیم و عملیات اسکن کردن هر پروت از هر آی پی را با استفاده از لوپ های تو در تو به یک Goroutine اختصاص می دهیم

WaitGroup قبل از اجرای هر Goroutine باید یک واحد افزایش یابد

در آخر با waitGroup.Wait() صبر می کنیم تا تمام Goroutine ها اجرا شوند و بعد از آن برنامه بسته می شود

استفاده از Goroutine باعث می شود پردازش در چند رشته مختلف انجام شود در نتیجه سرعت دریافت پاسخ بالا می رود و به طور کلی این کار به بهینگی کد کمک می کند

## توضیح متد ها

ScanPort

```go
func ScanPort(IP string, port int, timeout time.Duration, waitGroup *sync.WaitGroup) {
    defer waitGroup.Done()

    var portAddress string = fmt.Sprintf("%s:%d", IP, port)
    var connection, error = net.DialTimeout("tcp", portAddress, timeout)

    if error != nil {
        fmt.Printf("[IP: %s: Port: %d] is close\n", IP, port)
        return
    }
    defer connection.Close()

    fmt.Printf("[IP: %s: Port: %d] is open\n", IP, port)

    if port == 21 || port == 25 || port == 587 {
        var banner string = BannerGrabbing(connection, port, timeout)
        fmt.Printf("[IP: %s: Port:%d] Bannar:\n%s\n", IP, port, banner)
    }
}
```

defer wg.Done() باعث می شود پس از به پایان رسیدن این Groutin شمارنده WaitGroup را کاهش یابد

net.DialTimeout با استفاده از این متد کانکشن را با پورت مورد نظر برقرار می کنیم این متد یک آدرس که شامل ای پی و شماره پروت مورد نظر است دریافت می کند (که این آدرس را با فرمت مورد نیاز در سطر قبلی تولید کردیم) ور در صورتی که اتصال برقرار نشود یک ارور برمیگداند

در شرط بعدی درصورتی که اروری وجود داشته باشد و اتصال با پورت برقرار نشده باشد آدرس ای پی و پورت مورد نظر به همراه پیغام مربوطه نمایش داده میشود

در غیر این صورت پیام باز بود پورت نمایش داده می شود

همچنین اگر پورتی که کانکشن با آن برقرار شده FTP(21), SMTP(587,25) باشد با استفاده از متد BannerGrabbing بنر مربوط به آن پورت را دریافت می کنیم و آن را نمایش می دهد

CreateIPRangr

```go
func CreateIPRange(startIP string, endIP string) []string {
    var startIpParts []string = strings.Split(startIP, ".")
    var endIpParts []string = strings.Split(endIP, ".")

    startYPart, _ := strconv.Atoi(startIpParts[2])
    endYPart, _ := strconv.Atoi(endIpParts[2])
    startXPart, _ := strconv.Atoi(startIpParts[3])
    endXPart, _ := strconv.Atoi(endIpParts[3])

    var ipRange []string = []string{}

    for i := startYPart; i <= endYPart; i++ {
        for j := startXPart; j <= endXPart; j++ {
            var IP string = fmt.Sprintf("%s.%s.%d.%d", startIpParts[0], startIpParts[1], i, j)
            ipRange = append(ipRange, IP)
        }
    }

    return ipRange
}
```

این تابع یک ای پی آغازین و پایانی را دریافت می کند و یک بازه از ای پی های میان آن دو را می سازد ابتدا ای پی ها با متد اسپلیت به چهار بخش تقسیم می شوند و درمرحله دوم بخش های سوم و چهارم (در اینجا به عنوان X , Y از این دو بخش یاد کردم )آیپی به صورت عددی در متغیر هایی ذخیره می شود تا در لوپ بعدی بتوان از آن استفاده کرد . ای پی های ساخته شده در یک آرایه ذخیره می شود و در نهایت متد آن آرایه را برمی گرداند

BannerGrabbing

```go
53  func BannerGrabbing(conection net.Conn, port int, timeout time.Duration) string {
54      conection.SetReadDeadline(time.Now().Add(timeout))
55
56      var buffer []byte = make([]byte, 4096)
57      var banner string
58
59      for {
60          response, error := conection.Read(buffer)
61
62          if error != nil {
63              break
64          }
65
66          banner += string(buffer[:response])
67      }
68
69      if port == 21 {
70          conection.Write([]byte("USER anonymous\r\n"))
71      } else if port == 25 || port == 587 {
72          conection.Write([]byte("EHLO example.com\r\n"))
73      }
74
75      for {
76          response, error := conection.Read(buffer)
77
78          if error != nil {
79              break
80          }
81
82          banner += string(buffer[:response])
83      }
84
85      if banner == "" {
86          return "No banner!!!"
87      }
88
89      return banner
90  }
```

این متد تلاش می کند بنر ها را از پورت های FTM, SMTP دریافت کند

در خط اول مشخص می کنیم متد حداکثر تا چه مدت زمانی برای دریافت اطلاعات منتظر بماند

متغیربافر برای نگهداری اطلاعاتی است که از سوی سرور ارسال می شود و بنر نیز همانطور که واضح است نسخه متنی بنر دریافت شده می باشد

با استفده از conection.read سعی می کنیم اطلاعاتی از پروت مورد نظر دریافت کنیم در صورتی که اروری وجود داشته باشد از حلقه خارج می شویم

در مرحله بعد با توجه به پورت مورد نظر درخواست هایی را ارسال می کنیم تا حداکثر اطلاعات ممکن را دریافت کنیم (برای FTP: user anonymous و برای SMTP: EHLO example.com)

سپس دوباره مرحله قبل را تکرا می کنیم و در صورتی که بنری دریافت شده باشد متد آن را برمیگرداند

# تست کد و مقایسه خروجی با Nmap

FTP

```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\USER\Desktop\پیالوم هژورپ\NetworkScanner> go run banner_grabber4.go
Enter start IP: 185.27.134.11
Enter end IP: 185.27.134.11
Enter timeout (seconds): 10
[IP: 185.27.134.11: Port: 22] is open
[IP: 185.27.134.11: Port: 80] is open
[IP: 185.27.134.11: Port: 587] is open
[IP: 185.27.134.11: Port: 25] is open
[IP: 185.27.134.11: Port: 21] is open
[IP: 185.27.134.11: Port:587] Bannar:
No banner!!!
[IP: 185.27.134.11: Port:25] Bannar:
No banner!!!
[IP: 185.27.134.11: Port: 443] is close
[IP: 185.27.134.11: Port:21] Bannar:
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 385 of 6900 allowed.
220-Local time is now 13:25. Server port: 21.
220-This is a private system - No anonymous login
220 You will be disconnected after 60 seconds of inactivity.

End!
PS C:\Users\USER\Desktop\پیالوم هژورپ\NetworkScanner>
```

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>nmap --script=banner -p 21 185.27.134.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-18 20:56 Iran Standard Time
Nmap scan report for 185.27.134.11
Host is up (0.093s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| banner: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------\x
|_0D\x0A220-You are user number 374 of 6900 allowed.\x0D\x0A220-Local ...

Nmap done: 1 IP address (1 host up) scanned in 1.74 seconds

C:\Users\USER>
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\USER\Desktop\پیالوم هژورپ\NetworkScanner> go run banner_grabber4.go
Enter start IP: 74.125.131.26
Enter end IP: 74.125.131.26
Enter timeout (seconds): 10
[IP: 74.125.131.26: Port: 21] is open
[IP: 74.125.131.26: Port: 22] is open
[IP: 74.125.131.26: Port: 80] is open
[IP: 74.125.131.26: Port: 587] is open
[IP: 74.125.131.26: Port: 443] is open
[IP: 74.125.131.26: Port: 25] is open
[IP: 74.125.131.26: Port:21] Bannar:
No banner!!!
[IP: 74.125.131.26: Port:587] Bannar:
No banner!!!
[IP: 74.125.131.26: Port:25] Bannar:
220 mx.google.com ESMTP 2adb3069b0e04-5451f13640dsi5188253e87.324 - gsmtp

End!
PS C:\Users\USER\Desktop\پیالوم هژورپ\NetworkScanner>
```

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>nmap --script=banner -p 25 74.125.131.26
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-18 21:01 Iran Standard Time
Nmap scan report for lu-in-f26.1e100.net (74.125.131.26)
Host is up (0.033s latency).

PORT   STATE SERVICE
25/tcp open  smtp
| banner: 220 mx.google.com ESMTP 38308e7fff4ca-3091fae9b53si41421301fa.1
|_82 - gsmtp

Nmap done: 1 IP address (1 host up) scanned in 0.94 seconds

C:\Users\USER>
```

```
No banner!!!
[IP: 185.199.108.153: Port:587] Bannar:
No banner!!!
[IP: 185.199.108.153: Port:21] Bannar:
No banner!!!
End!
PS C:\Users\USER\Desktop\بیالوم هژورپ\NetworkScanner> go run banner_grabber4.go
Enter start IP: 198.50.135.197
Enter end IP: 198.50.135.197
Enter timeout (seconds): 10
[IP: 198.50.135.197: Port: 21] is open
[IP: 198.50.135.197: Port: 587] is open
[IP: 198.50.135.197: Port: 443] is open
[IP: 198.50.135.197: Port: 25] is open
[IP: 198.50.135.197: Port: 80] is open
[IP: 198.50.135.197: Port: 22] is open
[IP: 198.50.135.197: Port:21] Bannar:
No banner!!!
[IP: 198.50.135.197: Port:25] Bannar:
220-ns75.panfletovirtual.com ESMTP Exim 4.96.2 #2 Tue, 18 Feb 2025 14:54:59 -0300
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.

[IP: 198.50.135.197: Port:587] Bannar:
220-ns75.panfletovirtual.com ESMTP Exim 4.96.2 #2 Tue, 18 Feb 2025 14:54:59 -0300
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.

End!
PS C:\Users\USER\Desktop\بیالوم هژورپ\NetworkScanner>
```

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>nmap --script=banner -p 21 198.50.135.197
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-18 21:26 Iran Standard Time
Nmap scan report for ns75.panfletovirtual.com (198.50.135.197)
Host is up (0.055s latency).

PORT   STATE SERVICE
21/tcp open  ftp

Nmap done: 1 IP address (1 host up) scanned in 4.15 seconds

C:\Users\USER>nmap --script=banner -p 25 198.50.135.197
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-18 21:26 Iran Standard Time
Nmap scan report for ns75.panfletovirtual.com (198.50.135.197)
Host is up (0.19s latency).

PORT   STATE SERVICE
25/tcp open  smtp
| banner: 220-ns75.panfletovirtual.com ESMTP Exim 4.96.2 #2 Tue, 18 Feb 2
|_025 14:56:52 -0300 \x0D\x0A220-We do not authorize the use of this s...

Nmap done: 1 IP address (1 host up) scanned in 1.21 seconds

C:\Users\USER>
```

تست بررسی بازه ای پی

```
PS C:\Users\USER\Desktop\بیالوم هژورپ\NetworkScanner> go run banner_grabber4.go
Enter start IP: 142.251.12.100
Enter end IP: 142.251.12.139
Enter timeout (seconds): 10
[IP: 142.251.12.102: Port: 25] is open
[IP: 142.251.12.112: Port: 25] is open
[IP: 142.251.12.101: Port: 80] is open
[IP: 142.251.12.101: Port: 443] is open
[IP: 142.251.12.100: Port: 21] is open
[IP: 142.251.12.102: Port: 587] is open
[IP: 142.251.12.100: Port: 587] is open
[IP: 142.251.12.104: Port: 587] is open
[IP: 142.251.12.103: Port: 21] is open
[IP: 142.251.12.100: Port: 25] is open
[IP: 142.251.12.107: Port: 22] is open
[IP: 142.251.12.105: Port: 21] is open
[IP: 142.251.12.105: Port: 443] is open
[IP: 142.251.12.104: Port: 443] is open
[IP: 142.251.12.118: Port: 25] is open
[IP: 142.251.12.109: Port: 587] is open
[IP: 142.251.12.107: Port: 80] is open
[IP: 142.251.12.103: Port: 25] is open
[IP: 142.251.12.102: Port: 22] is open
[IP: 142.251.12.105: Port: 587] is open
[IP: 142.251.12.100: Port: 22] is open
[IP: 142.251.12.105: Port: 22] is open
[IP: 142.251.12.101: Port: 22] is open
[IP: 142.251.12.118: Port: 22] is open
[IP: 142.251.12.101: Port: 21] is open
[IP: 142.251.12.105: Port: 25] is open
[IP: 142.251.12.108: Port: 22] is open
[IP: 142.251.12.112: Port: 80] is open
[IP: 142.251.12.104: Port: 25] is open
[IP: 142.251.12.106: Port: 25] is open
[IP: 142.251.12.102: Port: 80] is open
[IP: 142.251.12.102: Port: 443] is open
[IP: 142.251.12.110: Port: 25] is open
[IP: 142.251.12.107: Port: 25] is open
[IP: 142.251.12.107: Port: 443] is open
[IP: 142.251.12.112: Port: 587] is open
[IP: 142.251.12.100: Port: 80] is open
[IP: 142.251.12.112: Port: 443] is open
[IP: 142.251.12.109: Port: 21] is open
[IP: 142.251.12.107: Port: 21] is open
[IP: 142.251.12.131: Port: 22] is open
[IP: 142.251.12.103: Port: 22] is open
[IP: 142.251.12.109: Port: 25] is open
[IP: 142.251.12.108: Port: 21] is open
[IP: 142.251.12.108: Port: 443] is open
[IP: 142.251.12.127: Port: 80] is open
[IP: 142.251.12.103: Port: 587] is open
```

```
[IP: 142.251.12.100: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.112: Port:25] Bannar:
No banner!!!
[IP: 142.251.12.102: Port:587] Bannar:
No banner!!!
[IP: 142.251.12.100: Port:587] Bannar:
No banner!!!
[IP: 142.251.12.128: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.101: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.103: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.104: Port:25] Bannar:
No banner!!!
[IP: 142.251.12.112: Port:587] Bannar:
No banner!!!
[IP: 142.251.12.111: Port:587] Bannar:
No banner!!!
[IP: 142.251.12.106: Port:25] Bannar:
No banner!!!
[IP: 142.251.12.109: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.118: Port:25] Bannar:
No banner!!!
[IP: 142.251.12.104: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.106: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.105: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.110: Port:25] Bannar:
No banner!!!
[IP: 142.251.12.105: Port:587] Bannar:
No banner!!!
[IP: 142.251.12.133: Port:587] Bannar:
No banner!!!
[IP: 142.251.12.125: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.131: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.104: Port:587] Bannar:
No banner!!!
[IP: 142.251.12.107: Port:25] Bannar:
No banner!!!
[IP: 142.251.12.107: Port:21] Bannar:
No banner!!!
[IP: 142.251.12.101: Port:25] Bannar:
No banner!!!
[IP: 142.251.12.105: Port:25] Bannar:
```

```
[IP: 142.251.12.101: Port: 587] is close
[IP: 142.251.12.109: Port: 80] is close
[IP: 142.251.12.139: Port: 587] is close
[IP: 142.251.12.102: Port: 21] is close
[IP: 142.251.12.106: Port: 587] is close
[IP: 142.251.12.107: Port: 587] is close
[IP: 142.251.12.110: Port: 21] is close
[IP: 142.251.12.109: Port: 443] is close
[IP: 142.251.12.108: Port: 80] is close
[IP: 142.251.12.118: Port: 587] is close
[IP: 142.251.12.130: Port: 587] is close
[IP: 142.251.12.106: Port: 22] is close
[IP: 142.251.12.110: Port: 587] is close
[IP: 142.251.12.125: Port: 80] is close
[IP: 142.251.12.131: Port: 25] is close
[IP: 142.251.12.113: Port: 25] is close
[IP: 142.251.12.133: Port: 21] is close
[IP: 142.251.12.124: Port: 22] is close
[IP: 142.251.12.112: Port: 21] is close
[IP: 142.251.12.125: Port: 587] is close
[IP: 142.251.12.138: Port: 25] is close
[IP: 142.251.12.131: Port: 443] is close
[IP: 142.251.12.131: Port: 587] is close
[IP: 142.251.12.139: Port: 25] is close
[IP: 142.251.12.130: Port: 21] is close
[IP: 142.251.12.116: Port: 21] is close
[IP: 142.251.12.121: Port: 21] is close
[IP: 142.251.12.121: Port: 22] is close
[IP: 142.251.12.121: Port: 25] is close
[IP: 142.251.12.126: Port: 25] is close
[IP: 142.251.12.118: Port: 21] is close
[IP: 142.251.12.120: Port: 587] is close
[IP: 142.251.12.130: Port: 22] is close
[IP: 142.251.12.136: Port: 22] is close
[IP: 142.251.12.135: Port: 22] is close
[IP: 142.251.12.136: Port: 25] is close
[IP: 142.251.12.109: Port:587] Bannar:
220 smtp.gmail.com ESMTP 98e67ed59e1d1-2fc2d6d8ee3sm8828333a91.49 - gsmtp

[IP: 142.251.12.109: Port:25] Bannar:
220 smtp.gmail.com ESMTP d2e1a72fcca58-7324254685csm10183442b3a.4 - gsmtp

[IP: 142.251.12.108: Port:587] Bannar:
220 smtp.gmail.com ESMTP d9443c01a7336-220d5364676sm92211575ad.82 - gsmtp

[IP: 142.251.12.108: Port:25] Bannar:
220 smtp.gmail.com ESMTP 41be03b00d2f7-adb5813a6b5sm8149619a12.26 - gsmtp

End!
PS C:\Users\USER\Desktop\بیالوم هژورپ\NetworkScanner>
```

در نمونه های بالا Nmap فقط پورتی که انتظار می رود بنر ارسال کنند اسکن شده است اما در کد بنده چند پورت اسکن شده است