

بخش 1:

1-1-1:

Scanning در امنیت به فرآیند شناسایی آسیب‌پذیری‌ها و نقاط ضعف در یک سیستم یا شبکه رایانه‌ای از طریق بررسی نقص‌های امنیتی شناخته شده آن اشاره دارد. این معمولاً با استفاده از ابزارهای نرم‌افزاری تخصصی معروف به اسکنرهای آسیب‌پذیری انجام می‌شود که به طور سیستماتیک اجزای مختلف سیستم را تجزیه و تحلیل می‌کند تا اطلاعات زیر را از سیستم بدست بیاورد:

1. پورت‌های باز و خدمات: اسکن می‌تواند تشخیص دهد که کدام پورت‌ها در یک شبکه باز هستند و کدام سرویس‌ها با آن پورت‌ها مرتبط هستند. این اطلاعات مهم است زیرا درگاه‌های باز ممکن است آسیب‌پذیری‌هایی را نشان دهند که می‌توان از آنها سوء استفاده کرد.

2. سیستم عامل و نسخه‌های نرم‌افزار: اسکن می‌تواند به شناسایی سیستم عامل و نرم‌افزار نصب شده بر روی سرورها، ایستگاه‌های کاری و سایر دارایی‌های IT کمک کند. این اطلاعات برای تعیین اینکه آیا نسخه‌های نرم‌افزار آسیب‌پذیری شناخته‌شده‌ای دارند و نیاز به به‌روزرسانی دارند، مفید است.

3. خطاهای پیکربندی: اسکن می‌تواند خطاها یا پیکربندی‌های نادرست را در شبکه، مانند رمزهای عبور ضعیف، نام‌های کاربری و رمزهای عبور پیش‌فرض، یا پورت‌های ناامن شناسایی کند.

4. آلودگی به بدافزار: اسکن همچنین می‌تواند با شناسایی وجود فایل‌های مخرب و فعالیت مشکوک شبکه، هر گونه آلودگی بدافزار را در شبکه شناسایی کند.

اسکن بخش مهمی از ارزیابی‌های امنیتی منظم است و برای حفظ یکپارچگی و ایمنی سیستم‌ها و شبکه‌های کامپیوتری ضروری است.

2-1-1:

Footprinting در امنیت به فرآیند جمع‌آوری اطلاعات در مورد یک سیستم یا شبکه هدف با هدف شناسایی آسیب‌پذیری‌ها و نقاط ضعف احتمالی اشاره دارد. این اطلاعات می‌تواند توسط مهاجمان برای انجام حملات هدفمند علیه سیستم یا شبکه مورد استفاده قرار گیرد. **Footprinting** می‌تواند شامل تکنیک‌های مختلفی باشد،

از جمله تکنیک‌های غیرفعال مانند جستجوی آنلاین، بررسی اطلاعات در دسترس عموم، و استفاده از تاکتیک‌های مهندسی اجتماعی برای استخراج اطلاعات از کارکنان یا پیمانکاران. همچنین می‌تواند شامل تکنیک‌های فعال‌تر، مانند نقشه‌برداری شبکه و اسکن پورت، برای شناسایی دستگاه‌ها و سرویس‌های مختلف موجود در شبکه هدف باشد. Footprinting بخش مهمی از مرحله شناسایی یک حمله سایبری است و اغلب توسط مهاجمان برای جمع‌آوری اطلاعاتی که می‌تواند در برنامه ریزی و اجرای مراحل بعدی حمله مفید باشد، استفاده می‌شود.

در اصل، Footprinting شامل جمع‌آوری اطلاعات در مورد سیستم هدف است، در حالی که Scanning شامل کاوش فعال هدف برای یافتن نقاط ضعف است. هر دو تکنیک در مراحل اولیه ارزیابی امنیتی یا تست نفوذ مهم هستند و می‌توانند به شناسایی آسیب‌پذیری‌های بالقوه‌ای که ممکن است در طول یک حمله هدفمند مورد سوء استفاده قرار گیرند، کمک کنند.

3-1-1:

1. استفاده از فایروال: فایروال‌ها یکی از موثرترین راه‌ها برای جلوگیری از دسترسی غیرمجاز به شبکه شما هستند. آنها با مسدود کردن ترافیک ورودی از منابع غیرمجاز کار می‌کنند و از دسترسی آنها به شبکه شما جلوگیری می‌کنند.

2. از رمزهای عبور امن استفاده کنید: مطمئن شوید که از رمزهای عبور قوی و منحصر به فرد برای دستگاه‌های شبکه خود، از جمله روترها، سوئیچ‌ها و سرورها استفاده می‌کنید. گذرواژه‌ها باید حداقل 12 تا 16 کاراکتر داشته باشند و ترکیبی از حروف بزرگ و کوچک، اعداد و کاراکترهای خاص را شامل شود.

3. از رمزگذاری استفاده کنید: شما باید از رمزگذاری برای محافظت از داده‌های شبکه خود در حین حرکت از طریق اینترنت یا سایر شبکه‌ها استفاده کنید. این می‌تواند شامل استفاده از HTTPS برای ترافیک وب یا VPN برای دسترسی از راه دور باشد.

4. محدود کردن دسترسی: فقط اجازه دسترسی به شبکه خود را از منابع مطمئن بدهید و دسترسی به منابع حساس شبکه را برای پرسنل مجاز محدود کنید.

5. نرم افزار را به روز نگه دارید: مطمئن شوید که همه نرم افزارها و سیستم عامل‌ها را به روز نگه دارید تا از آسیب‌پذیری‌هایی که مهاجمان می‌توانند از آنها سوء استفاده کنند جلوگیری کنید.

در این قسمت از کتابخانه های زیر استفاده شده است:

```
import argparse
import inspect
import socket
import os
import ipaddress
```

-1

این قسمت از یک تابع استفاده شده که آدرس شروع ip و آدرس پایان ip و subnetmask را از کاربر گرفته و تک تک ip های در این محدوده را پینگ گرفته و اگر قابل دسترسی باشد، پیامی به کاربر نشان داده و گزارش را در فایل تکست در درایو C ذخیره می کند. برای کار با فرمت ip و subnetmask از کتابخانه ipaddress استفاده میکنیم. شبکه ای که برای اسکن از آن استفاده کردم، شبکه وای فای خوابگاه استفاده کردم با فرمت 172.24.71.0 و 255.255.255.0 subnetmask است. برای اجرا در cmd از قالب زیر استفاده میکنیم:

```
python Phase02-Part01.py scan_ip -start_ip "172.24.71.0" -end_ip "172.24.71.255 -subnetmask 24
```

که scan_ip نام تابع و start_ip و end_ip و subnetmask پارامترهای تابع هستند. نمونه برنامه اجرا شده:

```
IP range scan report for range from 172.24.71.0 to 172.24.71.50 with subnetmask = 24:
172.24.71.0 is reachable.
172.24.71.1 is reachable.
172.24.71.2 is reachable.
172.24.71.3 is reachable.
172.24.71.4 is reachable.
172.24.71.5 is reachable.
172.24.71.6 is reachable.
172.24.71.7 is reachable.
172.24.71.8 is reachable.
172.24.71.9 is reachable.
172.24.71.10 is reachable.
172.24.71.11 is reachable.
172.24.71.12 is reachable.
172.24.71.13 is reachable.
172.24.71.14 is reachable.
172.24.71.15 is reachable.
172.24.71.16 is reachable.
```

نمونه ذخیره گزارش در فایل :

Part01-Report.txt - Notepad

File Edit Format View Help

```
IP range scan report for range from 172.24.71.0 to 172.24.71.50 with subnetmask = 24:
172.24.71.0 is reachable.
172.24.71.1 is reachable.
172.24.71.2 is reachable.
172.24.71.3 is reachable.
172.24.71.4 is reachable.
172.24.71.5 is reachable.
172.24.71.6 is reachable.
172.24.71.7 is reachable.
172.24.71.8 is reachable.
172.24.71.9 is reachable.
172.24.71.10 is reachable.
172.24.71.11 is reachable.
172.24.71.12 is reachable.
172.24.71.13 is reachable.
172.24.71.14 is reachable.
172.24.71.15 is reachable.
172.24.71.16 is reachable.
172.24.71.17 is reachable.
172.24.71.18 is reachable.
172.24.71.19 is reachable.
172.24.71.20 is reachable.
172.24.71.21 is reachable.
172.24.71.22 is reachable.
172.24.71.23 is reachable.
172.24.71.24 is reachable.
172.24.71.25 is reachable.
172.24.71.26 is reachable.
172.24.71.27 is reachable.
172.24.71.28 is reachable.
172.24.71.29 is reachable.
172.24.71.30 is reachable.
172.24.71.31 is reachable.
172.24.71.32 is reachable.
```

در این قسمت شماره پورت شروع و شماره پورت پایان و آدرس ip مقصد و پروتکل برقراری ارتباط که tcp یا udp است را به عنوان ورودی به تابع می‌دهیم و برای هر پورت با توجه به نوع پروتکل تعیین شده، سوکت می‌سازیم و به پورت مربوطه وصل می‌کنیم که اگر وصل شود یعنی که پورت باز است. برای اجرا در cmd از قالب زیر استفاده می‌کنیم:

```
python Phase02-Part01.py get_open_port -start_port 1 -end_port 100 -
target_ip "172.24.71.177" -protocol "tcp"
```

که get_open_port نام تابع و start_port و end_port و target_ip و protocol پارامترهای تابع هستند. نمونه برنامه اجرا شده برای پروتکل tcp:

```
arg_spec = inspect.getargspec(args.func)
Port range scan from 1 to 100 in ip address = 172.24.71.177 by protocol = tcp report:
Port 1 with Protocol tcp is closed
Port 2 with Protocol tcp is closed
Port 3 with Protocol tcp is closed
Port 4 with Protocol tcp is closed
Port 5 with Protocol tcp is closed
Port 6 with Protocol tcp is closed
Port 7 with Protocol tcp is closed
Port 8 with Protocol tcp is closed
Port 9 with Protocol tcp is closed
Port 10 with Protocol tcp is closed
Port 11 with Protocol tcp is closed
Port 12 with Protocol tcp is closed
Port 13 with Protocol tcp is closed
Port 14 with Protocol tcp is closed
Port 15 with Protocol tcp is closed
Port 16 with Protocol tcp is closed
Port 17 with Protocol tcp is closed
Port 18 with Protocol tcp is closed
Port 19 with Protocol tcp is closed
Port 20 with Protocol tcp is closed
Port 21 with Protocol tcp is closed
Port 22 with Protocol tcp is closed
Port 23 with Protocol tcp is closed
Port 24 with Protocol tcp is closed
Port 25 with Protocol tcp is closed
Port 26 with Protocol tcp is closed
Port 27 with Protocol tcp is closed
Port 28 with Protocol tcp is closed
Port 29 with Protocol tcp is closed
Port 30 with Protocol tcp is closed
```

```
Port 68 with Protocol tcp is closed
Port 69 with Protocol tcp is closed
Port 70 with Protocol tcp is closed
Port 71 with Protocol tcp is closed
Port 72 with Protocol tcp is closed
Port 73 with Protocol tcp is closed
Port 74 with Protocol tcp is closed
Port 75 with Protocol tcp is closed
Port 76 with Protocol tcp is closed
Port 77 with Protocol tcp is closed
Port 78 with Protocol tcp is closed
Port 79 with Protocol tcp is closed
Port 80 with Protocol tcp is open
Port 81 with Protocol tcp is closed
Port 82 with Protocol tcp is closed
Port 83 with Protocol tcp is closed
Port 84 with Protocol tcp is closed
Port 85 with Protocol tcp is closed
Port 86 with Protocol tcp is closed
Port 87 with Protocol tcp is closed
Port 88 with Protocol tcp is closed
Port 89 with Protocol tcp is closed
Port 90 with Protocol tcp is closed
Port 91 with Protocol tcp is closed
Port 92 with Protocol tcp is closed
Port 93 with Protocol tcp is closed
Port 94 with Protocol tcp is closed
Port 95 with Protocol tcp is closed
Port 96 with Protocol tcp is closed
Port 97 with Protocol tcp is closed
Port 98 with Protocol tcp is closed
Port 99 with Protocol tcp is closed
Port 100 with Protocol tcp is closed
```

C:\Users\MMNazari1380\PycharmProjects\SecurityCourseProject02>

نمونه ذخیره گزارش در فایل :

Part01-Report.txt - Notepad

```
File Edit Format View Help

Port range scan from 1 to 100 in ip address = 172.24.71.177 by protocol = tcp report:
Port 1 with Protocol tcp is closed
Port 2 with Protocol tcp is closed
Port 3 with Protocol tcp is closed
Port 4 with Protocol tcp is closed
Port 5 with Protocol tcp is closed
Port 6 with Protocol tcp is closed
Port 7 with Protocol tcp is closed
Port 8 with Protocol tcp is closed
Port 9 with Protocol tcp is closed
Port 10 with Protocol tcp is closed
Port 11 with Protocol tcp is closed
Port 12 with Protocol tcp is closed
Port 13 with Protocol tcp is closed
Port 14 with Protocol tcp is closed
Port 15 with Protocol tcp is closed
Port 16 with Protocol tcp is closed
Port 17 with Protocol tcp is closed
Port 18 with Protocol tcp is closed
Port 19 with Protocol tcp is closed
Port 20 with Protocol tcp is closed
Port 21 with Protocol tcp is closed
Port 22 with Protocol tcp is closed
Port 23 with Protocol tcp is closed
Port 24 with Protocol tcp is closed
Port 25 with Protocol tcp is closed
Port 26 with Protocol tcp is closed
Port 27 with Protocol tcp is closed
Port 28 with Protocol tcp is closed
Port 29 with Protocol tcp is closed
Port 30 with Protocol tcp is closed
Port 31 with Protocol tcp is closed
Port 32 with Protocol tcp is closed
Port 33 with Protocol tcp is closed
Port 34 with Protocol tcp is closed
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

نمونه برنامه اجرا شده برای پروتکل udp:

```
arg_spec = inspect.getargspec(args.func)
Port range scan from 1 to 100 in ip address = 172.24.71.177 by protocol = udp report:
Port 1 with Protocol udp is open
Port 2 with Protocol udp is open
Port 3 with Protocol udp is open
Port 4 with Protocol udp is open
Port 5 with Protocol udp is open
Port 6 with Protocol udp is open
Port 7 with Protocol udp is open
Port 8 with Protocol udp is open
Port 9 with Protocol udp is open
Port 10 with Protocol udp is open
Port 11 with Protocol udp is open
Port 12 with Protocol udp is open
Port 13 with Protocol udp is open
Port 14 with Protocol udp is open
Port 15 with Protocol udp is open
Port 16 with Protocol udp is open
Port 17 with Protocol udp is open
Port 18 with Protocol udp is open
Port 19 with Protocol udp is open
Port 20 with Protocol udp is open
Port 21 with Protocol udp is open
Port 22 with Protocol udp is open
Port 23 with Protocol udp is open
Port 24 with Protocol udp is open
Port 25 with Protocol udp is open
Port 26 with Protocol udp is open
Port 27 with Protocol udp is open
Port 28 with Protocol udp is open
Port 29 with Protocol udp is open
Port 30 with Protocol udp is open
Port 31 with Protocol udp is open
Port 32 with Protocol udp is open
Port 33 with Protocol udp is open
Port 34 with Protocol udp is open
Port 35 with Protocol udp is open
```

نمونه ذخیره گزارش در فایل :

Part01-Report.txt - Notepad

File Edit Format View Help

```
Port range scan from 1 to 100 in ip address = 172.24.71.177 by protocol = udp report:
Port 1 with Protocol udp is open
Port 2 with Protocol udp is open
Port 3 with Protocol udp is open
Port 4 with Protocol udp is open
Port 5 with Protocol udp is open
Port 6 with Protocol udp is open
Port 7 with Protocol udp is open
Port 8 with Protocol udp is open
Port 9 with Protocol udp is open
Port 10 with Protocol udp is open
Port 11 with Protocol udp is open
Port 12 with Protocol udp is open
Port 13 with Protocol udp is open
Port 14 with Protocol udp is open
Port 15 with Protocol udp is open
Port 16 with Protocol udp is open
Port 17 with Protocol udp is open
Port 18 with Protocol udp is open
Port 19 with Protocol udp is open
Port 20 with Protocol udp is open
Port 21 with Protocol udp is open
Port 22 with Protocol udp is open
Port 23 with Protocol udp is open
Port 24 with Protocol udp is open
Port 25 with Protocol udp is open
Port 26 with Protocol udp is open
Port 27 with Protocol udp is open
Port 28 with Protocol udp is open
Port 29 with Protocol udp is open
Port 30 with Protocol udp is open
Port 31 with Protocol udp is open
Port 32 with Protocol udp is open
Port 33 with Protocol udp is open
Port 34 with Protocol udp is open
<
```

در این قسمت آدرس ip ماشین مقصد و شماره پورت و پروتکل استفاده شده را از کاربر میگیریم و به آدرس و پورت سوکت داده شده را اتصال میدهیم و با دستور

```
'netstat -ano | findstr :{} | findstr LISTENING'
```

باز بودن پورت داده شده را چک میکنیم و در صورت باز بودن با دستور

```
'tasklist /fi "pid eq {}" /fo csv /nh'
```


اطلاعات پردازش های در حال اجرای بروی پورت داده شده را به دست می آوریم. برای اجرا در cmd از قالب زیر استفاده میکنیم:

```
python Phase02-Part01.py get_running_services -ip_address  
"172.24.71.177" -port 80
```

که get_services نام تابع و ip_address و port و protocol پارامترهای تابع هستند. نمونه برنامه اجرا شده بر روی پورت 80 و پروتکل tcp:

```
arg_spec = inspect.getargspec(args.func)  
Services for ip address = 172.24.71.177 with port number = 80 and protocol = tcp report:  
service found : System  
service found : System
```

نمونه ذخیره گزارش در فایل :

 Part01-Report.txt - Notepad
File Edit Format View Help

```
Services for ip address = 172.24.71.177 with port number = 80 and protocol = tcp report:  
service found : System  
service found : System
```


نمونه برنامه اجرا شده بر روی پورت 1 و پروتکل udp:

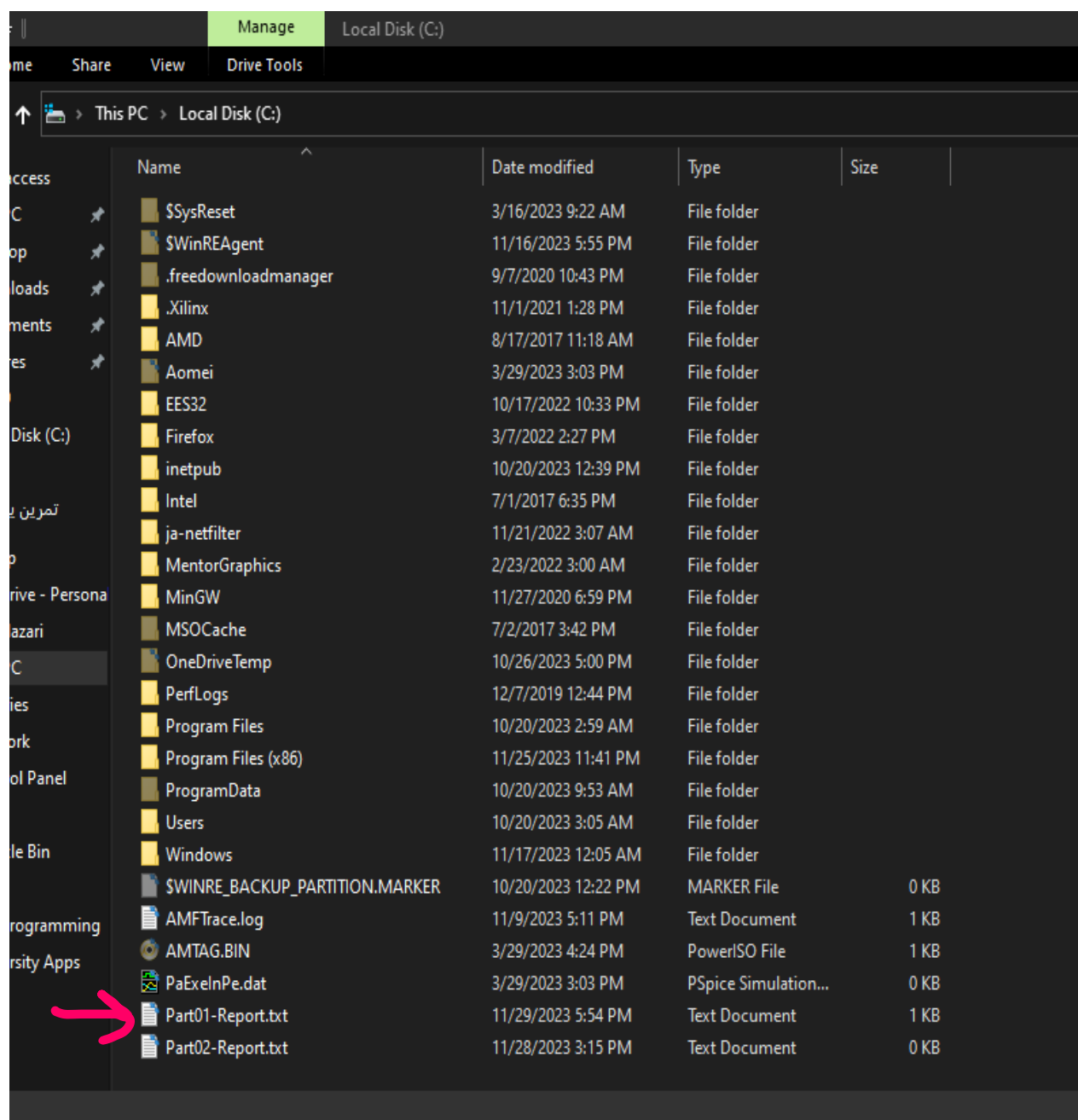
```
arg_spec = inspect.getargspec(args.func)
Services for ip address = 172.24.71.177 with port number = 1 and protocol = udp report:
service found : svchost.exe
service found : sqlservr.exe
service found : System
service found : System
service found : System
service found : System
service found : svchost.exe
service found : sqlservr.exe
service found : sqlservr.exe
C:\Users\MMNazari1380\Desktop\Projects\SecurityCourseProject01>
```

نمونه ذخیره گزارش در فایل :

Part01-Report.txt - Notepad
File Edit Format View Help

```
Services for ip address = 172.24.71.177 with port number = 1 and protocol = udp report:
service found : svchost.exe
service found : sqlservr.exe
service found : System
service found : System
service found : System
service found : System
service found : svchost.exe
service found : sqlservr.exe
service found : sqlservr.exe
```

محل ذخیره فایل های گزارش پروژه:



بخش 2:

2-1-1:

-sS:

این اسکن TCP Syn است که اسکن پیش فرض مورد استفاده Nmap است. یک بسته TCP SYN را به پورت هدف می فرستد تا مشخص کند که آیا باز، بسته یا فیلتر شده است.

-sT:

این اسکن اتصال TCP است که تلاش می کند یک اتصال کامل TCP با پورت هدف برقرار کند. قابل اطمینان تر از اسکن SYN است، اما همچنین توسط سیستم های تشخیص نفوذ به راحتی قابل تشخیص است.

-sV:

این سوئیچ تشخیص نسخه را فعال می کند، که سعی می کند نسخه سرویس در حال اجرا در پورت هدف را تعیین کند. این کار را با ارسال پروب های مختلف و تجزیه و تحلیل پاسخ ها انجام می دهد. این می تواند برای شناسایی آسیب پذیری های شناخته شده یا برای انگشت نگاری هدف برای کمک به شمارش یا بهره برداری بیشتر مفید باشد.

-sU:

این سوئیچ برای تعیین اسکن پورت UDP استفاده می شود. به طور پیش فرض، Nmap فقط پورت های TCP را اسکن می کند، اما با گزینه -sU، هر دو پورت TCP و UDP را اسکن می کند. یک پروتکل بدون اتصال است و اسکن آن می تواند دشوارتر باشد زیرا مانند TCP از دست دادن سه طرفه استفاده نمی کند. علاوه بر این، همه پورت های UDP به یک درخواست پاسخ نمی دهند، حتی اگر باز باشند.

2-1-2:

-F:

این یک مخفف برای گزینه "--fast" است که به nmap می گوید با استفاده از پروب های کمتری نسبت به اسکن معمولی اسکن سریع انجام دهد. این می تواند اسکن را سرعت بخشد، اما ممکن است برخی از پورت ها یا خدمات را نیز از دست بدهد.

-O:

این مخفف گزینه "osscan-guess--" است که به nmap می گوید سعی کند سیستم عامل هر میزبان را بر اساس پاسخ هایش به پروب های مختلف حدس بزند. این می تواند برای شناسایی انواع سیستم ها در یک شبکه مفید باشد.

-A:

این مختصر مجموعه ای از گزینه ها است که اسکن تهاجمی را امکان پذیر می کند. به طور خاص، این شامل سوئیچ های "osscan-guess--"، "--version-all"، "--traceroute" و "script all--" است. این می تواند گزینه مفیدی برای اسکن دقیق تر باشد، اما ممکن است در برخی از سیستم ها هشدارهای امنیتی را ایجاد کند.

:3-1-2

-sn: Disable port scanning. Host discovery only.

-pn: Disable host discovery. Port scan only.

:2-2

در این قسمت از کتابخانه های زیر استفاده میکنیم:

```
import argparse
import inspect
import nmap
```

-1

در این قسمت از portScanner و متد scan کتابخانه nmap استفاده میکنیم و صرفا محدوده ip بصورت a.b.c.d-e یعنی از a.b.c.d تا a.b.c.e را بصورت ورودی به تابع میدهیم. برای اجرا در cmd از قالب زیر استفاده میکنیم:

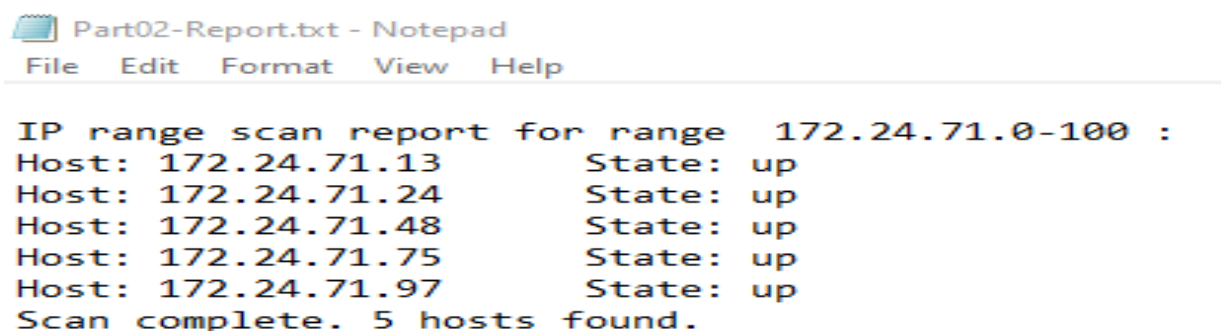
```
python Phase02-Part02.py scan_ip -ip_range "172.24.71.0-100"
```

که scan_ip نام تابع و ip_range پارامتر تابع است. نمونه برنامه اجرا شده:

```
arg_spec = inspect.getargspec(args.func)
IP range scan report for range 172.24.71.0-100 :
Host: 172.24.71.13      State: up
Host: 172.24.71.24      State: up
Host: 172.24.71.48      State: up
Host: 172.24.71.75      State: up
Host: 172.24.71.97      State: up
Scan complete. 5 hosts found.

C:\Users\MMNazari1380\PycharmProjects\SecurityCoursePro
```

نمونه ذخیره گزارش در فایل :



Part02-Report.txt - Notepad

File Edit Format View Help

```
IP range scan report for range 172.24.71.0-100 :
Host: 172.24.71.13      State: up
Host: 172.24.71.24      State: up
Host: 172.24.71.48      State: up
Host: 172.24.71.75      State: up
Host: 172.24.71.97      State: up
Scan complete. 5 hosts found.
```

-2

در این قسمت آدرس ip مقصد و محدوده پورت را به صورت $a-b$ یعنی از a تا b به تابع می‌دهیم و با استفاده از اسکنر در nmap ، state هر پورت را به ازای پروتکل های مختلف بدست می آوریم. برای اجرا در cmd از قالب زیر استفاده میکنیم:

```
python Phase02-Part02.py scan_port -target_ip "172.24.71.177" -port_range "1-100"
```

که scan_port نام تابع و target_ip و port_range پارامترهای تابع هستند. نمونه برنامه اجرا شده بر روی محدوده پورت 1-100:

```
arg_spec = inspect.getargspec(args.func)
Port range scan from range 1-100 in ip address = 172.24.71.13 report:
Protocol: tcp
Port: 53      State: closed
```

نمونه ذخیره گزارش در فایل :

Part02-Report.txt - Notepad
File Edit Format View Help

```
Port range scan from range 1-100 in ip address = 172.24.71.24 report:
Protocol: tcp
Port: 53      State: closed
```

-3

در این قسمت آدرس ip مقصد و پروتکل پورت مقصد و محدوده آن را به تابع می‌دهیم و با استفاده از اسکنر و ویژگی های فیلد پورت ، سرویس درحال اجرا بر روی آن پورت را بدست میاوریم. برای اجرا در cmd از قالب زیر استفاده میکنیم:

```
python Phase02-Part02.py scan_service -target_ip "172.24.71.177" -
protocol "tcp" -port_range "1-100"
```

که scan_service نام تابع و target_ip و port_range و protocol پارامترهای تابع هستند. نمونه برنامه اجرا شده بر روی محدوده پورت 1-100 و پروتکل tcp:

```
Services for ip address = 172.24.71.177 with protocol = tcp and port range = 1-100 report:  
Port : 80/http open  
C:\Users\MMN\azani1380\Documents\SecurityCourse\Project02\
```

مشاهده میشود که سرویس http در حال اجرا بر روی این پورت است.

نمونه ذخیره گزارش در فایل :

Part02-Report.txt - Notepad
File Edit Format View Help

```
Services for ip address = 172.24.71.177 with protocol = tcp and port range = 1-100 report:  
Port : 80/http open
```

-4

محل ذخیره فایل های گزارش پروژه:

