دانشکده‌ی مهندسی کامپیوتر

# شبکه‌های کامپیوتری

**نیم‌سال دوم ۰۴-۱۴۰۳**

استاد: امیرمهدی صادق‌زاده

تمرین سوم

پاسخ‌دهنده: معین آعلی - ۴۰۱۱۰۵۵۶۱

**فهرست مسائل**

## پاسخ مسئله‌ی ۱.

### آ

پردازه‌های A و B به عنوان کلاینت عمل می‌کنند و از پورت‌های موقت استفاده می‌کنند. (Ephemeral-Port) اما پردازه C که در این سناریو به عنوان سرور HTTP است، که باید با یک پورت ثابت کار کند. چون صورت سوال اشاره کرده است که در پروتوکل ارتباطی ما HTTP است، پس پورت پردازه C برابر با ۸۰ است.

### ب

در حالت عادی، این امکان وجود ندارد. چون که پورت و آیپی با هم Socket-Address را تشکیل می‌دهند و باید یک کلید یکتا باشد. پس اگر یک پردازه بخواهد روی یک پورت خاص bind کند، سیستم‌عامل آن پورت را برای این پردازه رزرو می‌کند و دیگر اجازه نمی‌دهد که پردازه دیگری روی آن پورت bind کند. اما حالات استثنا و خاص هم وجود دارد، مانند:

- فرض کنیم یک هاست دارای چند آیپی است. می‌توان یک پردازه روی یک آیپی و پورت و پردازه دیگر روی همان پورت و آیپی دیگر گوش کند.
- با استفاده از قابلیت SO-REUSEPORT می‌توان کاری کرد که دو پردازه روی یک پورت گوش دهند و سیستم‌عامل بین آن‌ها load-balancing انجام دهد.
- و موارد دیگر...

### ج

با اینکه در لایه‌ی لینک روش‌هایی برای تشخیص خطا وجود دارد، اما این روش‌ها فقط خطاهای local را در هر لینک مجزا بین دو گره تشخیص می‌دهند. در مسیر یک بسته از مبدا تا مقصد، بسته از چندین لینک و روتر عبور می‌کند و خطاهایی ممکن است در بافر روترها یا حافظه‌ی آن‌ها رخ دهد. پس برای اطمینان از اینکه بسته به برنامه‌ی مقصد به درستی رسیده است، نیاز است در لایه‌ی انتقال یک checksum داشته باشیم.

## پاسخ مسئله‌ی ۲.

- UDP ساده‌تر از TCP است (بدون کنترل اتصال، شماره‌گذاری بسته‌ها یا کنترل ازدحام)، پس سرعت انتقال داده بیشتر می‌شود.
- نیاز به حداقل تأخیر در برنامه‌هایی که نیاز دارند داده خیلی سریع برسد. مثل تماس صوتی یا لایو استریم. تأخیر کم مهم‌تر از تحویل تضمینی همه داده‌ها است. TCP به خاطر مکانیسم‌های تصحیح خطا، تأخیر بیشتری دارد که برای این نوع اپلیکیشن‌ها مناسب نیست.
- این اپلیکیشن‌ها طوری طراحی شده‌اند که حتی اگر برخی بسته‌ها از دست برود، همچنان به کار ادامه می‌دهند (مثلاً در تماس صوتی، یک کلمه گم شود ولی مکالمه قطع نشود).

برای مثال می‌توان به برنامه تماس تصویری و تماس صوتی و بازی‌های انلاین اشاره کرد.

## پاسخ مسئله‌ی ۳.

**آ**

**ب**

**ج**

## پاسخ مسئله‌ی ۴.

**آ**

**ب**

**ج**

## پاسخ مسئله‌ی ۵.

### آ

این دستور قصد دارد آدرس آیپی متناظر با دامنه sharif.edu را پیدا کند. با این تفاوت که از یک DNS سرور با آدرس 4.2.2.4 استفاده می‌کند نه از DNS سرور پیش‌فرض سیستم.

پاسخ داده شده دارای اطلاعات زیر هست:

- آدرس آیپی ورژن ۴ متناظر با این دامنه
- مقدار TTL مربوط به این آدرس آیپی . این مقدار برابر ۸ است در این درخواست.
- زمان پاسخ سرور DNS برابر با ۲۲۱ msec است.
- پاسخ روی پورت ۵۳ و پروتوکل UDP ارسال شده است.
- زمان اجرای دستور. البته احتمالا در این مورد زمان ست نشده بوده و در جواب زمان Unix-Timestamp برگردانده شده است.
- اندازه‌ی پیام ارسال شده از سمت سرور که در این جواب ۵۵ بایت است.
- عبارت Truncated یعنی پاسخ DNS خیلی بزرگ بوده، داخل UDP جا نشده، و dig فقط بخشی از آن را نشان داده است.

### ب

رکورد MX مشخص می‌کند که ایمیل‌هایی که به این دامنه ارسال می‌شوند، باید به کدام سرورهای ایمیل تحویل داده شوند. این درخواست از DNS سرور نوشته شده درخواست می‌کند تا رکوردهای MX دامنه‌ی sharif.edu را پیدا کند.

پاسخ داده شده دارای اطلاعات زیر هست:

- این دامنه دارای دو رکورد MX است.
- این رکوردها هردو دارای اولویت ۱۰ هستند.
- این رکوردها دارای TTL برابر ۶۰ هستند.
- باقی اطلاعات مشابه با بخش قبل هستند.

وقتی ایمیلی ارسال می‌شود، سرور فرستنده تلاش می‌کند ایمیل را به سروری با کمترین عدد اولویت بفرستد. اگر چند سرور اولویت یکسان داشته باشند، یکی از آن‌ها انتخاب می‌شود (اغلب به طور تصادفی یا بر اساس load-balancing ).

## پاسخ مسئله‌ی ۶.

در ادامه هر یک از روش‌ها را جدا توضیح می‌دهیم:

### TCP-Scanning

در این روش، اسکنر تلاش می‌کند اتصال کامل TCP را با پورت مقصد برقرار کند. یعنی ابتدا بسته‌ی SYN ارسال می‌شود، اگر پورت باز باشد SYN-ACK برمی‌گردد، و سپس ACK ارسال می‌شود تا اتصال کامل شود.
مزایا:

- بسیار دقیق است. وقتی ارتباط کامل برقرار شد، کاملاً مطمئن هستید که پورت باز است.

معایب:

- بسیار راحت توسط سیستم مقصد شناسایی می‌شود (در لاگ‌ها ثبت می‌شود).
- نسبت به روش‌های دیگر کندتر است.
- به راحتی فایروال جلوی آن را می‌تواند بگیرد

### SYN-Scanning

در این روش، فقط بسته‌ی SYN ارسال می‌شود. اگر پورت باز باشد، SYN-ACK پاسخ داده می‌شود، ولی اسکنر دیگر ACK نمی‌فرستد و اتصال را نیمه‌کاره رها می‌کند.
مزایا:

- سریع‌تر از Scanning TCP است.
- در لاگ‌های سیستم هدف کمتر دیده می‌شود.

معایب:

- ممکن است توسط فایروال‌ها یا IDS/IPS ها مسدود شود.

### FIN-Scanning

در این روش، بسته‌ی FIN به پورت مقصد ارسال می‌شود. اگر پورت بسته باشد، دستگاه پاسخ RST می‌دهد. اگر باز باشد، هیچ پاسخی نمی‌دهد (بر اساس استاندارد .(TCP
مزایا:

- می‌تواند از بعضی فایروال‌ها عبور کند

معایب:

- روی سیستم‌های ویندوز اغلب کار نمی‌کند، چون ویندوز به بسته FIN پاسخ RST می‌دهد حتی اگر پورت باز باشد.
- قابل اطمینان نیست مگر در سیستم‌های خاص ( مثل Unix/Linux ).

### UDP-Scanning

در این روش، بسته UDP به پورت ارسال می‌شود. اگر پورت بسته باشد، معمولا پیام ICMP-Port-Unreachable برمی‌گردد. اگر باز باشد، معمولا پاسخی دریافت نمی‌شود.
مزایا:

- تنها روش کاربردی برای شناسایی پورت‌های باز UDP

معایب:

- سخت‌ترین روش برای تشخیص وضعیت واقعی (باز یا بسته).
- کند است چون نیاز به تایم‌اوت دارد.
- ممکن است توسط فایروال‌ها مسدود شود.

**جمع‌بندی موارد:**

- TCP دقیق و کند و آشکار
- SYN سریع و نیمه مخفی
- FIN مخفی‌تر ولی کمتر قابل اطمینان
- UDP تنها راه برای بررسی UDP ولی کند و غیرقابل پیش‌بینی

سوال عملی
توضیحات نحوه کار کردن nmap

**پاسخ مسئله‌ی ۷.**