به نام خدا

درس شبکه‌های کامپیوتری

پروژه

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۰۲ـ۰۳

استاد:

دکتر کامبیز میزانیان

طراحان پروژه:

عرشیا اخوان، امیرمهدی نامجو و هومان کشوری

فهرست

نکات قابل توجه

۱. مهلت تحویل این پروژه تا تاریخ ۳۱ تیر ۱۴۰۳ است.

۲. برای تحویل پروژه، باید یک ویدیو بین ۵ تا ۱۵ دقیقه‌ای ضبط کرده و به طور کامل مراحل اجرای پروژه و درستی کارکرد آن را نشان داده و همچنین کدها و مراحل فنی مد نظر برای انجام آن را توضیح بدهید. در موارد استثنا، ممکن است از بعضی افراد پروژه به صورت مجازی و آنلاین نیز تحویل گرفته شود.

۳. پروژه به صورت تک‌نفری بوده و هر گونه کپی‌برداری از کد دیگران تقلب محسوب می‌شود. هم‌فکری بین دانشجویان و همچنین استفاده از منابع اینترنتی و LLMها تنها به شرطی که تسلط کامل روی تک تک بخش‌های پروژه و کد وجود داشته باشد، امکان پذیر است. عدم تسلط بر روی هر بخش از پروژه که تحویل داده شده باشد به دلایلی نظیر استفاده از ChatGPT و... مصداق تقلب محسوب می‌شود.

۴. این پروژه به لحاظ حجم کد مد نظر، نسبتا کوچک است اما به دانش قابل‌توجهی از شبکه (به خصوص پروتکل‌های TCP، UDP، DNS و IP) نیاز دارد. به همین دلیل توصیه می‌شود پروژه را به روزهای آخر موکول نکنید تا در صورت ابهام در مفاهیم آن، زمان کافی برای یادگیری یا پرسیدن ابهامات از دستیاران آموزشی وجود داشته باشد.

مقدمه

همان طور که می‌دانید، در بسیاری از شبکه‌ها (چه به صورت کشوری و چه شبکه‌های محلی شرکت‌ها) محدودیت‌هایی اعمال می‌شود که امکان دسترسی به اینترنت آزاد را از کاربران سلب می‌کند. با این حال از طریق VPN ها می‌توان بسیاری از این محدودیت‌ها را دور زد و به اینترنت دسترسی پیدا کرد. در این حالت، عملا کامپیوتری در شبکه وجود دارد که به اینترنت آزاد دسترسی دارد و شما هم می‌توانید به آن کامپیوتر (سرور) دسترسی پیدا کنید و از این طریق، آن کامپیوتر نقش واسط بین شما و اینترنت را ایفا می‌کند و پیام‌های شما ابتدا برای آن ارسال شده و از طرف آن برای محیط خارجی ارسال شده و سپس جواب آن به واسطه آن سرور برای شما ارسال می‌شود.
با این وجود روش‌هایی از طریق تحلیل الگوی ترافیک اینترنت وجود دارد که به کنترل‌کنندگان شبکه کمک می‌کند تا سرورهای VPN را تشخیص بدهند و آن‌ها را از کار بیندازند. به عنوان مثال در صورتی که ترافیک ورودی و خروجی یک سرور تقریبا برابر باشد یا الگوهای خاصی از پیام‌های TCP و UDP در آن‌ها رد و بدل بشود، می‌توان متوجه شد که احتمالا از این سرور به عنوان VPN استفاده می‌شود و جلوی آن را گرفت. با این وجود، یکسری از پروتکل‌های اساسی که برای اینترنت مورد استفاده هستند نظیر پروتکل DNS قابل بلاک کردن نیستند و در نتیجه می‌توان به نحوی از آن‌ها برای دور زدن محدودیت‌ها استفاده کرد.
هدف در این پروژه پیاده‌سازی سرور و کلاینتی است که از طریق DNS بتوانند محدودیت‌های اعمال شده بر شبکه را دور بزنند.

پروژه

محیط مورد بررسی

محیط مورد بررسی در این پروژه را می‌توان به دو بخش دنیای بیرون و خارج شبکه محدود شده تقسیم کرد. دنیای خارج شبکه محدود شده همان اینترنت آزاد است و می‌تواند شامل هر سایتی باشد. در داخل شبکه محدود شده، تمامی کامپیوترهایی که داریم به جز یکی از آن‌ها به لحاظ ارتباط با خارج از شبکه محدود هستند و امکان برقراری ارتباط با بیرون را ندارند. تنها یک کامپیوتر وجود دارد که امکان ارتباط با شبکه آزاد خارجی را دارد ولی این کامپیوتر در شبکه داخلی تنها از طریق پروتکل DNS قابل دسترس است و سایر پروتکل‌های آن بسته شده‌اند. توجه کنید که بسته‌های DNS در قالب UDP در لایه Transport منتقل می‌شوند.

پیاده‌سازی

خواسته اصلی این پروژه این است که شما کد کلاینت و سروری پیاده سازی کنید که بتواند محدودیت این شبکه را دور بزند. به طور دقیق‌تر کد کلاینت باید روی هر کدام از کامپیوترهای داخلی این شبکه قابلیت اجرا داشته باشد و وظیفه آن اتصال به کامپیوتری است که در این شبکه امکان ارتباط با بیرون را دارد. کد سرور روی این کامپیوتر خاص اجرا می‌شود و ظیفه دارد درخواست‌های کلاینت‌ها را به بیرون از شبکه منتقل کرده و جواب آن را به کلاینت برگرداند.

برای این موضوع، باید به نکات مختلفی توجه کنید. همان طور که گفته شد، امکان ارتباط با این کامپیوتر تنها از طریق پروتکل DNS امکان پذیر است. به طور کلی DNS تنها امکان انتقال ۵۱۲ بایت داده را دارد در حالی که پیام‌های TCP می‌توانند تا ۶۴ کیلوبایت هم باشند. برای رفع این مشکل می‌توان در قسمت Additional در DNS، یک OPT Record از نوع Unknown اضافه کرد و در RData آن بسته‌های TCP مربوطه را قرار دارد. بدین ترتیب و از طریق استفاده از مکانیزم EDNS امکان انتقال ۴ کیلوبایت پیام در یک بسته DNS امکان‌پذیر خواهد شد. به بیان دگیر شما باید عملا درخواست‌های دسترسی به اینترنت که عموما درخواست‌هایی از جنس TCP هستند را در قسمت RDATA در بسته‌های DNS به سرور ارسال کنید و سرور نیز بعد از بررسی پیام با پروتکل TCP با بیرون ارتباط برقرار کرده و جواب آن را به صورت DNS به کلاینت بازگرداند.

در این میان باید توجه کنید که MTU در شبکه عموما حدود ۱۵۰۰ بایت است و در نتیجه باید MSS پکت‌های TCP ای که در RDATA قرار می‌دهید را طوری تنظیم کنید که این مسئله را مد نظر قرار بدهد.

به علاوه توجه کنید که احتمالا برای این که بسته‌های لایه Transport را بتوان به صورت مجزا از اینترفیس عادی شبکه بررسی کرد، احتمالا باید یک Interface جدید تعریف کنید. این کار در لینوکس از طریق TUN/TAP API خود کرنل و در ویندوز از طریق ابزارهایی نظیر Wintun امکان پذیر است. با این وجود به دلیل وجود Documentation بیش‌تر توصیه می‌شود از TUN/TAP API لینوکس استفاده کنید. همچنین برای هندل کردن ارتباط اینترفیس جدید ایجاد شده با اینترفیس عادی اینترنت موجود، می‌توانید یک NAT در سرور تعریف کنید.

موارد تحویلی

موارد زیر برای این پروژه مطلوب است:

- کد کامل پیاده‌سازی کلاینت و سرور توضیح داده شده در قسمت‌های قبلی. این کد می‌تواند با هر یک از زبان‌های مطرح نظیر C، C++، Python، Java، Go، Rust

و... یا ترکیبی از این زبان‌ها نوشته شده باشد.

- مستند دو تا ده صفحه‌ای از کلیات کار انجام شده و چالش‌هایی که با آن مواجه شده‌اید.
- ویدیو ۵ تا ۱۵ دقیقه‌ای از توضیح پروژه و اجرای آن

منابع مفید

مطالعه زیر می‌تواند در انجام این پروژه به شما کمک کنند:

- RFC 2671: Extension Mechanisms for DNS (EDNS0): https://datatracker.ietf.org/doc/html/rfc2671
- RFC 1035: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION: https://datatracker.ietf.org/doc/html/rfc1035
- RFC 793: TRANSMISSION CONTROL PROTOCOL: https://datatracker.ietf.org/doc/html/rfc793
- RFC 879: The TCP Maximum Segment Size and Related Topics: https://datatracker.ietf.org/doc/html/rfc879