بسمه تعالی

پروژه درس طراحی الگوریتم
تطبیق رشته

«ویروس‌ها به شهر حمله کرده‌اند، باید چاره‌ای اندیشید!»

بقیه قصه‌اش را خودتان در ذهن هالیوودی‌تان بسازید.

تعدادی فایل باینری در اختیار شما قرار گرفته است. در مجموع دو دسته فایل وجود دارد:

فایلهای آلوده (Malware) و فایل‌های بی خطر (Benign).

هر فایل به صورت مجموعه‌ای از بایتها است که هر ۴ bit آن معادل یک کاراکتر HEX است.

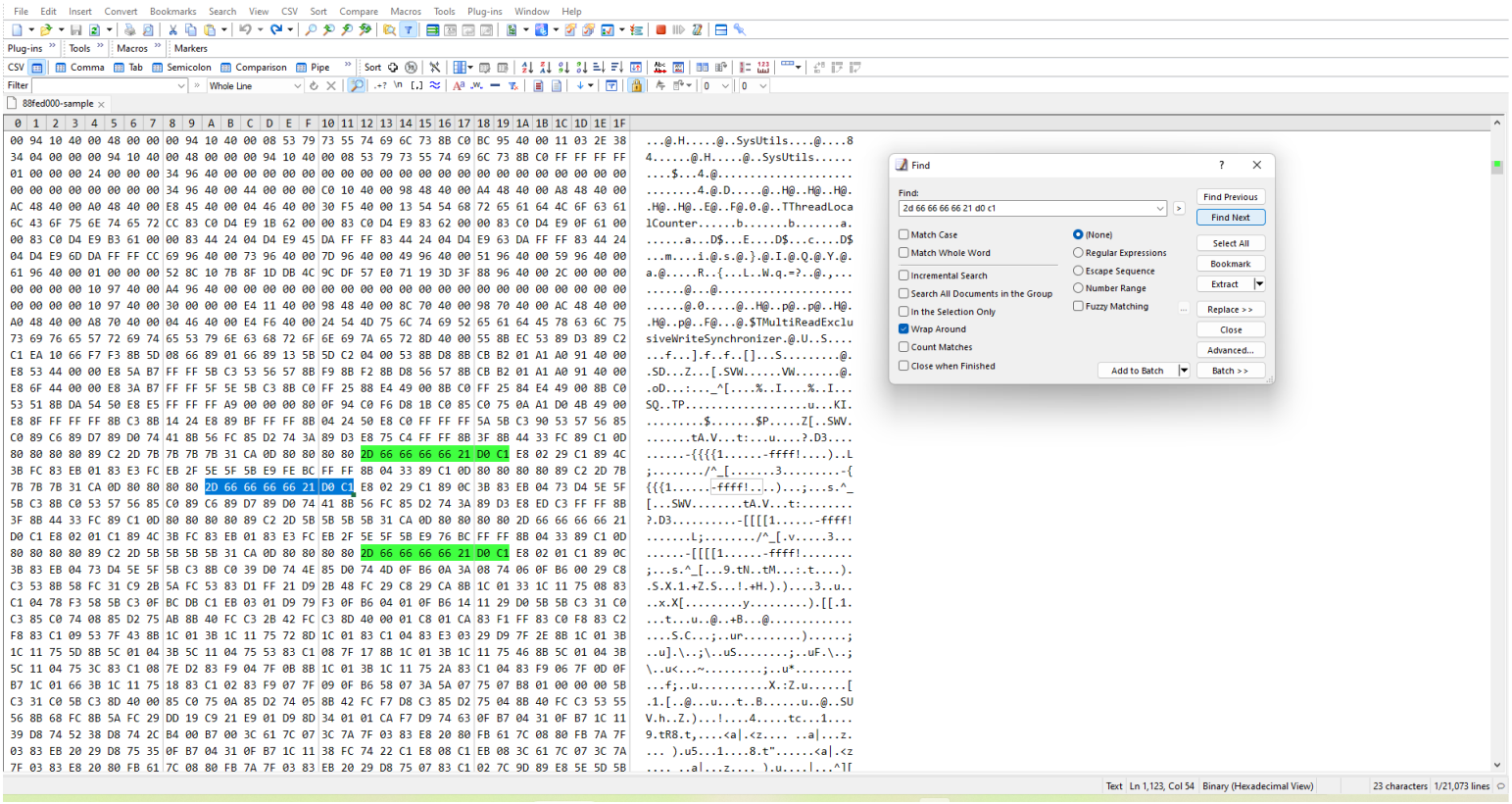
یعنی مجموعه کاراکترهای 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F که معادل ارقام صفر تا ۱۵ هستند.

به این ترتیب هر فایل باینری در کامپیوتر معادل متنی با کاراکترهای HEX است و مثلا فایلی به اندازه دقیقا ۱۷ بایت معادل متنی به طول ۳۴ کاراکتر خواهد بود.

شما می‌توانید با نرم افزاری مانند EmEditor یک فایل باینری را به صورت باینری باز کنید:

گزینه open as binary (Hexadecimal view) را انتخاب کنید.

مثال:



در این پروژه قرار است الگوهایی در متن پیدا کنید که با کمک تطبیق آنها با فایل‌های ورودی بتوان فایل بدافزار (آلوده) را از فایل بی خطر تشخیص داد. هر الگو می‌تواند یک زیررشته ساده، یا ترکیبی از زیررشته‌های مختلف به اشکال مختلف باشد.

الگوها برای تطبیق بدافزار طراحی می‌شوند، یعنی هدفشان یافتن بدافزار است و پیش فرض آن است که فایل ورودی بی خطر است مگر آنکه با یکی از الگوهای موجود تطبیق پیدا کند. ضمنا برای بدافزار بودن، تطبیق با حداقل یک الگو لازم است (بیشتر از یک الگو هم ممکن است تطبیق پیدا کند ولی لزومی ندارد). اگر فایلی با هیچ الگویی تطبیق پیدا نکرد، یعنی بی خطر است.

مثلا الگوی زیر را ببینید:

P1: *,45A3874BCCB3625374,{18},FAC345FFB,*,DEE4528,{54-73},74672BCD34EA567F

این الگو یعنی، از ابتدای متن هر تعداد کاراکتر (صفر تا بی نهایت!) برو جلو، بعد باید زیر رشته پیوسته "45A3874BCCB3625374" را ببینی، بعد دقیقا ۱۸ کاراکتر برو جلو (هر کاراکتری که باشند)، بعد باید زیررشته "FAC345FFB" را ببینی، بعد هر تعداد کاراکتر که خواستی برو جلو تا به زیر رشته "DEE4528" برسی، بعد حداقل ۵۴ و حداکثر ۷۳ کاراکتر دیگر جلو برو و نهایتا الگوی "74672BCD34EA567F" را ببین. اگر این تطبیق از ابتدا تا انتها اتفاق افتاد، یعنی الگوی P1 با متن (فایل) ورودی تطبیق پیدا کرده است و لذا فایل، یک بدافزار است.

البته وحشت نکنید، الگوهای شما ممکن است به سادگی یک زیر رشته تک و تنهای ساده باشد. اشکالی ندارد و لزوما قرار است نیست الگوها این قدر چند تکه و دارای دنگ و فنگ باشند (شاید هم بعضا لازم شود!).

ضمنا ساختار فوق صرفا یک مثال است و اصلا لزومی ندارد الگویی که شما پیدا میکنید دقیقا همین فرمت را داشته باشد و می‌توانید به هر شکلی که مایلید الگوی دلخواه خودتان را پیدا کنید و به یک نحوی و با یک فرمت دلخواهی آن را بیان کنید.

مهم آن است که تا جای ممکن با کمترین تعداد الگو و با کمترین طول ممکن برای هر الگو، بیشترین تعداد فایل بدافزار تشخیص داده شود با رعایت این شرط بسیار مهم که حتی المقدور هیچ فایل بی خطری اشتباها به عنوان بدافزار شناسایی نشود (یا تعداد این گونه خطاها که به آن خطای مثبت کاذب (False Positive) می‌گویند، در حداقل ممکن باشد (مثلا زیر ۱۰ تا)).

مثلا شما می‌توانید برای هر فایل بدافزار، کل متن آن فایل را به عنوان الگوی اول در نظر بگیرید و برای سایر فایلها هم الگوهای دیگری به اندازه طول آنها در نظر بگیرید. در این صورت برای مثلا ۱۰۰۰۰ بدافزار تعداد ۱۰۰۰۰ الگو معرفی کرده اید (یعنی تعداد الگوها زیاد است) که طول هر یک نیز خیلی بلند (به اندازه کل متن) است. البته قطعا هیچ خطای مثبت کاذبی نخواهید داشت. این ورژن از کار، میتواند نقطه شروع باشد و سعی کنید تعداد و طول الگوها را کم کنید بدون آنکه خطای FP ایجاد کنید. یا میتوانید برای نقطه شروع حالتی را در نظر بگیرید که هیچ الگویی وجود ندارد، پس هیچ خطای مثبت کاذبی هم وجود نخواهد داشت، هرچند هیچ بدافزاری هم تشخیص داده نخواهد شد. حالا سعی کنید اولین الگویی که با تعداد از بدافزارها تطبیق دارد ولی با هیچ فایل بی خطری تطبیق ندارد را پیدا کنید. اگر موفق شدید، کافی است فایلهایی که با الگوی اول تطبیق یافتند را کنار بگذارید و این شیوه را تکرار کنید تا فایلهای بعدی با الگوی بعدی پیدا شوند و به همین ترتیب!

البته اینها صرفا پیشنهادهای اولیه است. شما آزادید از هر روشی استفاده کنید.

در این پروژه مجاز به استفاده از کتابخانه های آماده برای پیاده سازی بخش الگوریتم پروژه نیستید. برای لود کردن فایلها، ذخیره سازی نتایج، نمایش و ... طبیعتا میتوانید هرچیزی که خواستید include/import کنید، ولی برای خود الگوریتم، باید قدم به قدم آن را خودتان توسعه دهید. ممکن است روش موجودی را مطالعه کنید و بخواهید از آن برای بخشی از کارتان استفاده کنید (مثلا الگوریتم‌های پویا برای تطبیق بزرگترین زیر رشته مشترک – longest common substring matching). در این صورت باید آن را کامل پیاده کنید.

قرار نیست نتیجه کارتان خیلی دقت بالایی داشته باشد، خلاقیت و تلاش شما برای پیاده سازی ایده های مختلفی که میزنید (حتی اگر خیلی نتیجه بخش نباشد یا نتایجش خیلی دلچسب نباشد) میتواند بخش قابل توجهی از نمره پروژه را به دست آورد.

فایلهای پروژه به صورت بخش‌بندی شده به شما تحویل داده شده است. بخش Benign انواع مختلفی از فایلهای Benign را شامل میشود. بخش Malware به صورت دسته بندی شده ارایه شده است که هر بخش آن مربوط به یک نوع بدافزار خاص است که احتمال دارد دارای الگوی مشترکی باشند. بنابراین می‌توانید تلاش کنید برای هر گروه بدافزاری مجزا، الگو (الگوهای) مشترکی پیدا کنید. البته همه فایل‌های ارائه شده عقیم شده‌اند و امکان اجرا شدن ندارند، اما در فرآیند عقیم سازی، اطلاعات اصلی و مهم فایلها به هیچ وجه از بین نرفته است، بنابراین امکان یافتن الگوهای کارا و موثر وجود دارد.

خبر خوب: با تعداد حداکثر ۱۰۰ الگو، که هر یک طول حدودی زیر ۲۰۰ کاراکتر دارند، میتوان بدون خطای مثبت کاذب، همه بدافزارها را تشخیص داد. البته این فقط برای داشتن حدودی از وضعیت بهینه است و برای اینکه بدانید چقدر به یک جواب خوب نزدیک شده اید.

فرمت ورودی و خروجی:

هنگام تست برنامه، ورودی برنامه شما یک پوشه (Folder) است که دارای تعدادی فایل است (زیرفولدر وجود ندارد). برنامه شما تمام فایلهای داخل پوشه را اسکن میکند، و مواردی که به عنوان بدافزار شناسایی میشود را از داخل آن پوشه cut کرده و به پوشه دیگری با نام Malwares انتقال میدهد (اگر پوشه Malwares وجود ندارد، برنامه خودش آن را ایجاد کند).

مثلا ورودی زیر به برنامه داده میشود که دارای ۱۰۰ فایل است: C:\MyProject\Files

برنامه شما پوشه C:\MyProject\Files\Malwares را در صورت نیاز ایجاد کرده و تمام بدافزارهای تشخیص داده شده را به آن انتقال می‌دهد.

آنچه تحویل می‌دهید:

کدهای پروژه + گزارش چند صفحه ای از مراحل کار و الگوریتم توسعه داده شده و نتایج حاصل شده.

پروژه شما به صورت آنلاین تحویل گرفته خواهد شد. کدی که به LMS ارسال کرده اید را در روز تحویل پروژه دانلود کرده و همان کد را در سیستم خودتان اجرا خواهید کرد. پس دقت کنید کد آپلود شده بدون نقص باشد.

در گزارش پروژه، ضمن توضیح فعالیتها و الگوریتمهایی که توسعه داده اید، چالشهایی که برخورد کرده اید و راه حلهای آنها، نتایجی که حاصل کرده اید را نیز با دقت گزارش کنید.

الگوهایی که یافته اید (با هر الگوریتمی) را در گزارش خود ارائه کنید (اگر نیاز داشت، فرمت ارائه هر الگو را نیز معرفی کنید تا معنای الگوهای پیشنهادی فهمیده شود)

همچنین در گزارشتان تحلیلی از مرتبه زمانی الگوریتمتان ارائه کنید. این مرتبه زمانی بستگی به تعداد الگوها و طول هر یک خواهد داشت. مثلا اگر ۵۰۰ الگو با مجموع طول ۷۰۰۰۰۰ کاراکتر دارید، برای اسکن هر فایل ورودی ممکن است الگوریتم شما شامل تعداد عملیاتی برابر با ضریب ثابتی از ۷۰۰۰۰۰ باشد (اگر مرتبه الگوریتمتان بر حسب مجموع طول الگوها خطی باشد). یا ممکن است مرتبه چندجمله‌ای بر حسب تعداد الگو یا طول متوسط الگوها یا ... داشته باشید. این موضوع را دقیق تحلیل کنید و در گزارشتان یک بخش مجزا برای تحلیل زمان اجرایی الگوریتم داشته باشید.

بس است دیگر: شروع کنید ...