دانشکده مهندسی کامپیوتر	**باسمه تعالی** **فرم پیشنهاد پروژه کارشناسی**	دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

تحویل پیشنهاد پروژه به دانشکده و ثبت نهایی آن در پورتال: (این قسمت توسط کارشناسان آموزش دانشکده تکمیل می شود.)

تاریخ تحویل پیشنهاد پروژه به آموزش دانشکده:

تاریخ ثبت نهایی در پورتال آموزشی دانشگاه:

مشخصات دانشجو:

نام و نام خانوادگی: سید مجتبی غضنفری

شماره دانشجویی: ۴۰۰۳۱۰۳۵

رایانامه (ایمیل) دانشجو: sey.ghaz@aut.ac.ir

نیمسال و سال تحصیلی ثبت‌نام پروژه: نیمسال دوم از سال تحصیلی ۱۴۰۴

توضیح ۱: دانشجو موظف است حداکثر **دو ماه** پس از ثبت نام پروژه فرم تکمیل شده پیشنهاد پروژه را، که به امضای استاد راهنمای او رسیده است، به آموزش دانشکده تحویل دهد. انجام سر وقت این مرحله نشان دهنده بخشی از رعایت زمانبندی انجام پروژه توسط دانشجو است.

توضیح ۲: آموزش دانشکده پیشنهاد پروژه دریافتی را جهت تعیین داور و انجام داوری در اختیار گروه آموزشی استاد راهنمای دانشجو قرار می‌دهد. گروه‌های آموزشی **حداکثر طی دو ماه** داوری را انجام داده و در صورت تصویب در گروه، پیشنهاد پروژه را جهت تصویب در دانشکده و ثبت در پورتال آموزشی دانشگاه در اختیار آموزش دانشکده قرار می‌دهند. دانشجویان موظفند با داور(ان) پیشنهاد پروژه خود در ارتباط بوده و نظرات آنان را، با راهنمایی استاد راهنمای خود و در مهلت مقرر گروه برای تصویب پیشنهاد پروژه، بر روی پیشنهاد پروژه خود اعمال نمایند.

توضیح ۳: مهلت درج نمره پروژه دانشجویانی که در نیمسال اول یا در تابستان سال تحصیلی پروژه را اخذ نموده‌اند، **سی‌ام مهر** سال تحصیلی بعد و برای دانشجویانی که در نیمسال دوم پروژه را اخذ نموده‌اند، **سی و یکم‌ام فروردین** سال تحصیلی بعد است.

توضیح ۴: فاصله زمانی بین ثبت نهایی پیشنهاد پروژه (تصویب شده) در پورتال آموزشی دانشگاه و دفاع از پروژه **حداقل سه ماه** است و امکان دفاع قبل از سپری شدن این فاصله زمانی وجود ندارد. همچنین، دفاع از پروژه کارشناسی با اعلان عمومی و با حضور مخاطبان در حضور داوران انجام خواهد شد. لازم است دانشجویان حداقل **سه هفته** قبل از فرارسیدن مهلت درج نمره پروژه (توضیح ۳)، پایان‌نامه تایپ شده خود را، که به تأیید استاد راهنما رسیده است، در اختیار آموزش دانشکده و داور(ان) پروژه قرار داده و مقدمات برگزاری جلسه دفاع را، با هماهنگی آموزش دانشکده، فراهم آورند.

توضیح ۵: لازم است دانشجویان رویه دانشگاه صنعتی امیرکبیر با عنوان «چگونگی ثبت‌نام، تصویب، و دفاع از پایان‌نامه در مقطع کارشناسی» را که با شماره AUT-PR-3210 بر روی سایت معاونت آموزشی دانشگاه قرار گرفته است مطالعه کنند.

امضای دانشجو: **تاریخ: ۱۴۰۳/۱۲/۲۰**

استاد راهنمای پروژه:

نام و نام خانوادگی: **امضا:** **تاریخ:**

عنوان پروژه:

عنوان فارسی:

سامانه مدیریت دسترسی به پرونده‌ها بر اساس حساسیت محتوایی با استفاده از Fanotify در لینوکس

عنوان انگلیسی:

File Access Management Solution Based on Content Sensitivity Using Fanotify in Linux

داور(ان) پیشنهاد پروژه:

داور اول:

نام و نام خانوادگی: سید احمد جوادی **امضا:** **تاریخ:** ۱۴۰۴/۰۲/۲۷

داور دوم:

نام و نام خانوادگی: **امضا:** **تاریخ:**

توضیح: با امضای این قسمت داور(ان) محترم تأیید می‌کنند که

۱- دانشجو، با راهنمایی استاد راهنمای خود، اصلاحات مورد نظر داور(ان) را انجام داده و عنوان و محتوای پیشنهاد پروژه از نظر ایشان قابل قبول است.

۲- دانشجو با مفاهیم پیش‌نیاز و مهارت‌های ضروری و پایه انجام این پروژه آشنایی داشته یا کسب آن برای دانشجو در طول انجام پروژه امکان‌پذیر است.

۳- موارد زیر در پیشنهاد پروژه مورد توجه قرار گرفته است:

- عنوان پروژه به طور کامل و دقیق موضوع پروژه را نشان می‌دهد و محتوای پروژه با عنوان پروژه کاملاً مطابقت دارد.
- پیشنهاد پروژه شامل بخش‌های مقدمه، مرور پیشینه پژوهش، رویکرد پیشنهادی، روش ارزیابی، مراحل و زمان‌بندی انجام پروژه، امکانات لازم و لیست مراجع و منابع است.
- اجزای سامانه مورد نظر پروژه در یک نمودار بلوکی نشان داده شده و ورودی‌ها و خروجی‌های آن مشخص شده‌اند.
- تأکید پروژه بر روی مسائل عملی و علمی و مهارت‌های مهندسی کامپیوتر است و پروژه منجر به توسعه نرم‌افزار، سخت‌افزار یا ترکیبی از آن دو و با درجه سختی و حجم مناسب یک پروژه سه واحدی است.
- پروژه بر مبنای استفاده از دروس کارشناسی تعریف شده است.
- چنانچه قرار است در پروژه از ابزارها، نرم‌افزارها، یا محیط‌های آماده استفاده شود، این موارد با صراحت بیان شده و مشخص شده است چه بخش‌هایی و با چه مقداری تلاش سهم دانشجو است.
- پروژه علاوه بر بخش مطالعاتی-نظری، حدود ۱۵۰ ساعت کار عملی لازم داشته و انجام آن حداقل ۳ ماه زمان نیاز دارد.

تصویب پیشنهاد پروژه:

تصویب در گروه آموزشی:

نام و نام خانوادگی مدیر گروه: **امضا:** **تاریخ:**

تصویب در شورای آموزشی–پژوهشی دانشکده:

نام و نام خانوادگی معاون آموزشی: **امضا:** **تاریخ:**

تعریف پروژه: (دانشجو می‌تواند با اضافه کردن فاصله لازم بر روی پرونده قابل ویرایش این سند، توضیحات خود را در هر یک از قسمت‌های زیر تایپ کند.)

۱- مقدمه (بیان مسئله کاربردی، ضرورت، انگیزه، اهداف، و چالش‌های انجام این پروژه):

امنیت اطلاعات در سیستم‌عامل‌های لینوکس یکی از مسائل حیاتی در حفاظت از داده‌های حساس است. در بسیاری از مواقع، کنترل دسترسی به پرونده‌ها باید نه تنها بر اساس نقش‌های کاربری یا مجوزهای سطح سیستم‌عامل ، بلکه با توجه به محتوای پرونده‌ها و حساسیت آن‌ها نیز مدیریت شود. این حساسیت می‌تواند از عواملی مانند وجود کلمات خاص، عبارات مهم یا الگوهای خاص در متن پرونده ناشی شود. به این ترتیب، سیستم‌های امنیتی نیاز به راهکارهایی دارند که بتوانند محتویات پرونده‌ها را به‌طور هوشمندانه تحلیل کرده و بر اساس آن تصمیم‌گیری کنند. انواع مختلف پرونده‌های متنی مانند پرونده‌های پیکربندی، گزارشات، کدهای منبع، صفحات وب، پرونده‌های JSON و XML و بسیاری دیگر به‌طور معمول در سیستم‌های لینوکسی وجود دارند که می‌توانند شامل اطلاعات حساسی باشند که نیاز به حفاظت دارند.

در این پروژه، هدف پیاده‌سازی سامانه‌ای است که هنگام باز شدن یک پرونده، ابتدا محتوای آن پرونده را بررسی کرده و بر اساس وجود کلمات، عبارات مهم یا الگوهای خاص، حساسیت آن پرونده را به‌طور پویا محاسبه کند. به‌عنوان مثال، اگر محتوای پرونده شامل کلمات حساس مانند "رمز عبور"، "داده‌های مالی" یا حتی اطلاعات شخصی مانند نام و شماره شناسنامه باشد، حساسیت پرونده به میزان بالاتری تخصیص داده می‌شود. این حساسیت می‌تواند به‌صورت یک درصد یا درجه محرمانگی قابل تنظیم باشد. همچنین این سامانه امکان محافظت بر اساس هش پرونده‌ها را نیز در اختیار کاربران قرار می‌دهد.

پس از ارزیابی حساسیت محتوا، دسترسی به پرونده بر اساس این حساسیت و نقش‌های دسترسی تعیین خواهد شد. به‌این‌ترتیب، سامانه قادر خواهد بود تا به‌طور دقیق و هوشمندانه، دسترسی کاربران به پرونده‌ها را مدیریت کند و اجازه دهد یا ندهد که یک پرونده خاص بر اساس حساسیت محتویات یا موارد تعیین شده قبلی توسط مدیر سامانه برای کاربر باز شود.

زیرسامانه Fanotify در هسته لینوکس به‌عنوان یک زیرسامانه مفید برای نظارت بر تغییرات و دسترسی به پرونده‌ها در زمان واقعی، می‌تواند به‌طور مؤثر برای پیاده‌سازی این سرویس استفاده شود. این قابلیت اجازه می‌دهد که تغییرات محتویات پرونده در زمان باز شدن حساسیت آن پرونده محاسبه گردد. سپس سیستم، دسترسی به پرونده را بر اساس حساسیت محتوای آن و نقش‌های امنیتی پیش‌تعریف‌شده، مدیریت خواهد کرد.

اهداف پروژه:

- **طراحی سامانه‌ای جهت دریافت رویدادهای دسترسی به پرونده از سیستم‌عامل**
  این ماژول به عنوان نقطه‌ی شروع پردازش عمل می‌کند و وظیفه دارد تمام دسترسی‌های صورت‌گرفته به پرونده‌ها را پایش کند.
- **شناسایی مشخصات پرونده شامل مسیر، نوع، اندازه، هش و فرمت جهت آماده‌سازی برای ارزیابی.**
  اطلاعات پایه‌ی پرونده پس از شناسایی توسط سامانه در اختیار سایر بخش‌ها قرار می‌گیرد تا تحلیل انجام شود.
- **تحلیل اولیه پرونده با استفاده از هش یا مشخصات ساختاری برای تطبیق با سیاست‌های امنیتی.**
  اگر قوانین تعریف‌شده تنها بر اساس هش باشند، تحلیل محتوایی حذف شده و تصمیم‌گیری سریع انجام می‌پذیرد.
- **در صورت نیاز، استخراج محتوای متنی از پرونده‌هایی با فرمت PDF و DOCX برای بررسی محتوای داخلی.**
  سیستم دارای زیرساختی برای parsing پرونده‌های متنی و نیمه‌ساخت‌یافته است که در صورت لزوم فعال می‌شود.
- **تطبیق محتوای پرونده با سیاست‌های امنیتی شامل کلمات کلیدی و عبارات حساس.**
  محتوا با سیاست‌های فعال مقایسه شده و سطح حساسیت را مشخص می‌شود.
- **استفاده از مکانیزم کش برای ذخیره نتایج تحلیل و جلوگیری از پردازش تکراری پرونده‌های مشابه.**
  در صورتی که پرونده تغییری نکرده باشد، نتیجه تحلیل قبلی به صورت مستقیم استفاده می‌شود.
- **به‌کارگیری مکانیزمی برای تشخیص پرونده‌ها یا کاربران پرتکرار جهت عبور سریع از مرحله تحلیل.**
  این ماژول تحلیل‌های آماری انجام می‌دهد و رفتار سیستم را در بلندمدت بهینه می‌سازد.
- **اعمال فوری تصمیم امنیتی شامل مسدودسازی، ثبت لاگ یا ارسال گزارش.**
  با توجه به نتیجه و سیاست‌های سیستم، اقدام مناسب را انجام خواهد شد.
- **ارائه داشبورد مدیریتی برای مشاهده تصمیمات، تحلیل رویدادها، و پایش وضعیت سامانه.**

مدیر سیستم می‌تواند رفتار پرونده‌ها و کاربران را از طریق رابط گرافیکی کنترل و مدیریت کند.

چالش‌های پروژه:

- طراحی الگوریتمی برای تحلیل محتوای فایل‌ها با استفاده از کلمات کلیدی، عبارات خاص و الگوهای منظم، به گونه‌ای که ضمن حفظ دقت بالا، باعث کاهش محسوس در کارایی یا تأخیر در پاسخ‌گویی نشود.
- ایجاد زیرساختی برای تعریف و به‌روزرسانی سیاست‌های امنیتی بصورت پویا، بدون نیاز به توقف سامانه یا تداخل در عملکرد جاری آن.
- اطمینان از پایداری سامانه در برابر خطاهای تحلیل، تأخیر در پاسخ یا مشکلات ماژول‌ها، به‌گونه‌ای که مانع از تأثیر منفی بر عملکرد سیستم‌عامل شود.
- مدیریت کارآمد منابع سیستمی مانند حافظه و پردازنده، به‌ویژه هنگام تعامل هم‌زمان با چند فایل یا تعداد زیادی قوانین فعال، بدون تأثیرگذاری منفی بر سایر برنامه‌های در حال اجرا.
- امکان وقوع حلقه‌های بی‌پایان ورودی/خروجی (I/O loops) در صورت طراحی نادرست، به‌ویژه در مواردی که سامانه خود باعث تولید رویدادهای جدید در فایل سیستم می‌شود.
- کاهش اثربخشی کش در شرایطی که پرونده‌ها به طور مکرر تغییر می‌کنند یا کاربران پرتکرار با رفتار متغیر در سیستم فعال هستند.
- محدود بودن منابع مستند و رسمی در رابطه با Fanotify و برخی محدودیت‌های آن، که فرآیند توسعه، خطایابی و پشتیبانی فنی را دشوار می‌سازد.
- عدم امکان ارتباط مستقیم با فراخوانی‌های سیستمی مربوط به نوشتن (write) در سطح Fanotify باعث می‌شود تشخیص دقیق زمان تغییر واقعی محتوای فایل دشوار شود، و در نتیجه اعتبارسنجی اطلاعات ذخیره‌شده در کش با چالش مواجه گردد.
- محدود بودن ابزارها و کتابخانه‌های کارآمد برای استخراج محتوای متنی از فایل‌های پیچیده مانند PDF وWord ، به‌ویژه در زبان‌های سطح پایین‌تری مانند C یاRust .

۲- مروری بر پروژه‌ها و سامانه‌های مشابه و بیان نقاط قوتی که با انجام این پروژه حاصل می‌شود:

در زمینه کنترل دسترسی به پرونده‌ها و محافظت از داده‌ها، پروژه‌های مختلفی وجود دارند که عمدتاً بر اساس سیاست‌های ایستا یا ویژگی‌های ثابت مانند نام پرونده، هش یا ویژگی‌های فراداده پرونده‌ها عمل می‌کنند. در این بخش، به برخی از این سامانه‌ها اشاره می‌شود و پس از آن، نقاط قوت پروژه مورد بررسی قرار می‌گیرد.

پروژه‌های مشابه: SELinux یک فریم‌ورک امنیتی است که برای مدیریت دسترسی و محافظت از سیستم‌عامل لینوکس طراحی شده است. این پروژه به‌طور خاص برای اعمال سیاست‌های دسترسی در سطح هسته سیستم‌عامل به‌کار می‌رود و با استفاده از مدل‌های امنیتی مانند MAC، دسترسی به منابع سیستم را محدود می‌کند. SELinux با استفاده از سیاست‌های از پیش تعیین‌شده، به برنامه‌ها و کاربران اجازه می‌دهد تا دسترسی‌های خاصی به منابع سیستم مانند پرونده‌ها، دستگاه‌ها و فرآیند داشته باشند. در این سیستم، مدیریت دسترسی بیشتر بر اساس ویژگی‌های پرونده (مثل نام، مسیر و هش) می باشد. با این حال، SELinux به دلیل تمرکز بیشتر بر منابع سیستم و استفاده از سیاست‌های ایستا، قادر به تحلیل دقیق محتوای پرونده‌ها برای تعیین حساسیت آن‌ها نیست.

نقاط قوت پروژه پیشنهادی:

- کنترل دسترسی بر اساس محتوای پرونده: در پروژه‌های مشابه مانند SELinux، دسترسی به پرونده‌ها معمولاً بر اساس ویژگی‌های ایستا مانند نام پرونده، هش یا ویژگی‌های فراداده آن‌ها انجام می‌شود. در حالی که در پروژه پیشنهادی، محافظت از پرونده‌ها با تحلیل دقیق محتوای واقعی آن‌ها انجام می‌گیرد. این رویکرد به‌ویژه در محیط‌هایی که داده‌های حساس و تغییرپذیر وجود دارند، مانند اسناد قانونی، اطلاعات مالی، داده‌های پزشکی و غیره، کاربرد فراوانی دارد. به‌طور مثال، حتی اگر نام یک پرونده تغییر کند یا محتوای آن به‌طور قابل توجهی دستخوش تغییر شود، سیستم قادر خواهد بود حساسیت جدید پرونده را محاسبه کرده و دسترسی‌های مربوطه را به‌طور پویا تنظیم کند.
- ماژولار بودن و قابلیت گسترش پروژه: یکی دیگر از ویژگی‌های برجسته پروژه پیشنهادی، ماژولار بودن آن است. این ویژگی به‌طور قابل توجهی قابلیت گسترش سیستم را افزایش می‌دهد و امکان اضافه کردن تحلیلگر جدید برای انواع مختلف فرمت‌های پرونده‌ها (مانند PDF، XML، CSV و غیره) را فراهم می‌کند. با این امکان، در صورت نیاز به پردازش و محافظت از پرونده‌های جدید با ساختار خاص، می‌توان به‌راحتی تحلیلگرهای جدید را ایجاد و به سرویس اضافه کرد، بدون آن‌که نیازی به تغییرات عمده در ساختار اصلی سرویس باشد.
- بهینه‌سازی عملکرد از طریق کش و تحلیل رفتار کاربران/پرونده‌ها: سیستم پیشنهادی به‌گونه‌ای طراحی شده است که از مکانیزم کش برای ذخیره نتایج تصمیم‌گیری و تحلیل استفاده می‌کند تا از پردازش تکراری برای پرونده‌های تغییرنیافته جلوگیری شود. همچنین، یک لایه تحلیل رفتاری وجود دارد که کاربران یا پرونده‌های پرتکرار را شناسایی کرده و در صورت قابل اعتماد بودن، فرآیند ارزیابی را برای آن‌ها ساده‌سازی می‌کند. این طراحی علاوه بر افزایش سرعت سیستم، باعث کاهش مصرف منابع و بهبود مقیاس‌پذیری در بارهای بالا می‌شود.

۳- روش انجام پروژه (روش، نمودار بلوکی اجزای سامانه‌ی مورد نظر پروژه، ورودی‌ها و خروجی‌ها):

در پروژه پیشنهادی، هدف اصلی تحلیل و کنترل دسترسی به پرونده‌ها بر اساس محتوای آن‌ها است. برای این منظور، از یک سیستم ماژولار استفاده خواهد شد که قادر است به‌طور پویا محتوای پرونده‌ها را تحلیل کرده و بر اساس حساسیت محتوا، دسترسی‌ها را مدیریت کند. همچنین، این سامانه به‌وسیله Fanotify که یک API در سطح هسته سیستم‌عامل لینوکس است، به‌طور خودکار و به محض باز شدن پرونده‌ها فعال می‌شود و محتوای آن‌ها را برای تحلیل و بررسی کنترل دسترسی مورد استفاده قرار می‌دهد.

اجزای اصلی سامانه:

- **CoreEngine**

هسته مرکزی سیستم است که وظیفه راه‌اندازی تمامی مؤلفه‌های داخلی را بر عهده دارد و ارتباط میان آن‌ها را مدیریت می‌کند. همچنین، در زمان دریافت رویداد دسترسی از فایل‌سیستم، مسیر تحلیل را آغاز کرده و تصمیم‌گیری یا پردازش را به مؤلفه‌های مرتبط ارجاع می‌دهد.

- **AccessContextDB**

مرجع ذخیره‌سازی تمام سیاست‌های تعریف‌شده، نتایج کش شده، و تحلیل‌های رفتاری است. این پایگاه داده در زمان اجرا توسط RuleEvaluator و CacheManagerمورد استفاده قرار می‌گیرد و قابلیت به‌روزرسانی بدون نیاز به توقف سامانه را دارد.

- **CacheManager**

این بخش مسئول بررسی وجود داده‌ی کش‌شده برای فایل‌ها و بازیابی یا به‌روزرسانی آن‌ها است. در صورتی که نتیجه تحلیل قبلاً برای فایلی ذخیره شده باشد و اعتبار آن تأیید گردد، تحلیل مجدد انجام نمی‌شود و تصمیم مستقیم اعمال می‌گردد.

- **RuleEvaluator**

ماژول اصلی تصمیم‌گیری امنیتی است که مشخصات فایل، اطلاعات کاربر، هش، و در صورت نیاز محتوای استخراج‌شده را با سیاست‌های موجود تطبیق می‌دهد. در صورت لزوم، از ContentParser برای تحلیل دقیق‌تر استفاده می‌کند و نهایتاً نتیجه را به ActionHandler و CacheManager ارسال می‌نماید.

- **ContentParser**

در صورتی که قوانین نیازمند بررسی محتوای داخلی فایل باشند، این مؤلفه فعال شده و محتوای متنی فایل را استخراج می‌کند. پشتیبانی از فرمت‌هایی مانند PDF و Word به‌صورت ماژولار پیاده‌سازی شده و در صورت نیاز، قابل گسترش به سایر فرمت‌ها است.

- **AccessOptimizer**

رفتار کاربران و فایل‌ها را در طول زمان تحلیل می‌کند و موارد پرتکرار یا قابل‌اعتماد را شناسایی می‌نماید. این داده‌ها به‌عنوان ورودی کمکی به RuleEvaluator و CacheManager ارسال می‌شوند تا فرآیند تصمیم‌گیری بهینه‌تر و سریع‌تر انجام شود.

- **ActionHandler**

پس از تعیین حساسیت و تصمیم نهایی، این بخش وظیفه اجرای عملیاتی تصمیم را دارد؛ از جمله بلاک کردن دسترسی، ثبت لاگ، یا ارسال گزارش به داشبورد. طراحی آن به‌گونه‌ای است که در برابر خطا مقاوم بوده و از ایجاد اختلال در سامانه جلوگیری می‌کند.

- **InsightDashboard**

رابط گرافیکی سامانه برای مشاهده لاگ‌ها، گزارش‌ها و وضعیت تصمیمات امنیتی در لحظه است. مدیر سامانه می‌تواند از طریق این داشبورد سیاست‌ها را پایش، ارزیابی و در صورت نیاز تنظیم مجدد کند.

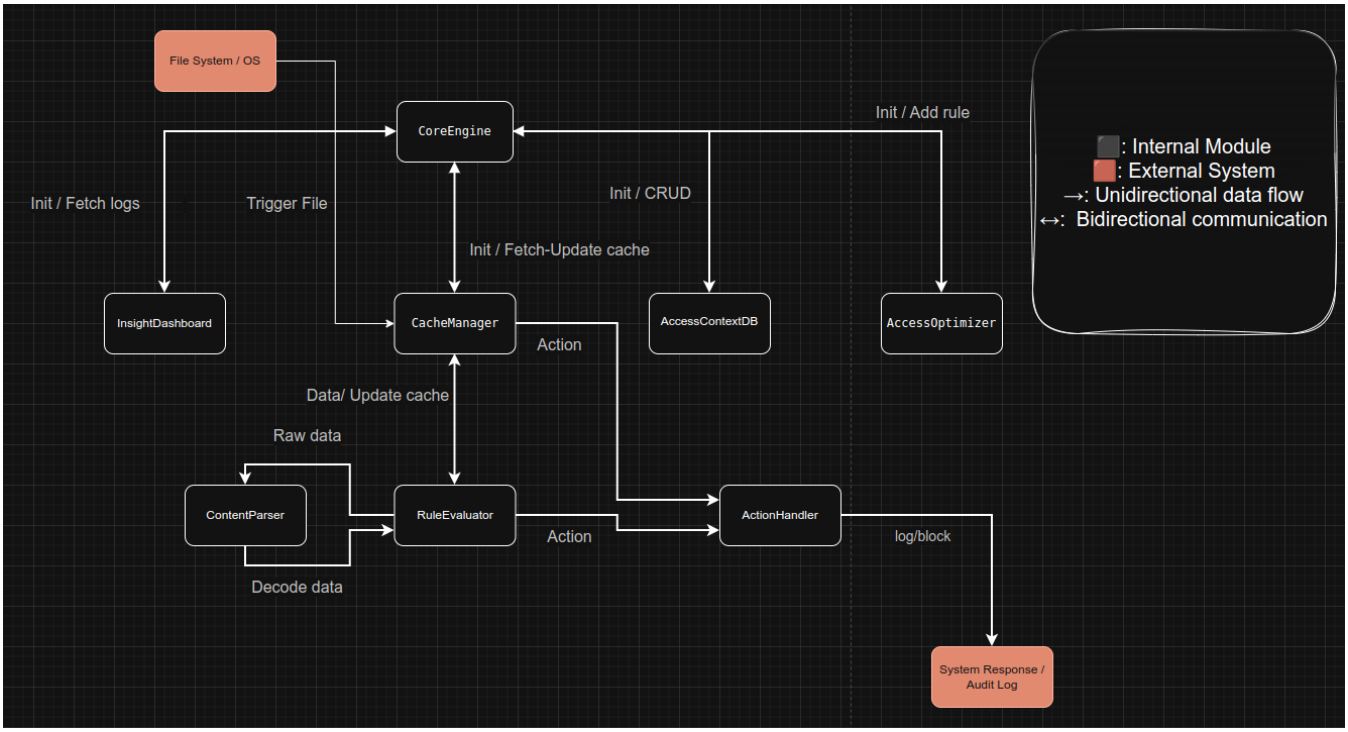
ورودی‌ها:

- پرونده‌هایی با فرمت‌های مختلف (مثلاً .txt، .pdf، .csv، .xml و غیره)
- قوانین و کش‌های مربوط به پرونده‌ها.
- عبارات منظم برای شناسایی محتوای حساس یا هش‌های از پیش تعیین شده.

خروجی‌ها:

- تصمیمات دسترسی برای هر کاربر و هر پرونده.
- گزارش‌های خطا در صورت وجود محتوای حساس که دسترسی به آن محدود شده است.

در ادامه نمودار بلوکی سامانه آورده شده است:



معادل فارسی اصطلاحات:

معادل فارسی	عبارت انگلیسی
سیستم فایل / سیستم‌عامل	File System / OS
موتور مرکزی	CoreEngine
سنسور دسترسی به فایل	File Access Sensor
داشبورد بینش (نمایش لاگ‌ها و گزارش‌ها)	InsightDashboard
پایگاه داده سیاست‌های دسترسی و کش	AccessContextDB
مدیر کش	CacheManager
ارزیاب قوانین	RuleEvaluator
تجزیه‌کننده محتوا	ContentParser
اجراکننده عملیات (بلاک، لاگ و...)	ActionHandler
بهینه‌ساز دسترسی	AccessOptimizer
پاسخ سیستم / لاگ ممیزی	System Response / Audit Log

جدول ۱: معادل فارسی اجزا سامانه

معادل فارسی	عبارت انگلیسی

Init / Fetch logs	مقداردهی اولیه / دریافت لاگ‌ها
Trigger File	تحریک فایل (رویداد دسترسی)
Init / Fetch-Update cache	مقداردهی اولیه / دریافت یا به‌روزرسانی کش
Init / CRUD	مقداردهی اولیه / عملیات ایجاد، خواندن و ...
Init / Add rule	مقداردهی اولیه / افزودن قاعده
Data / Update cache	داده / به‌روزرسانی کش
Raw data	داده خام
Decode data	محتوای تجزیه‌شده
Action	عملیات (مثلاً دستور بلاک یا لاگ)
log/block	لاگ‌گیری / بلاک‌کردن

جدول ۲: معادل فارسی برچسب‌های تعاملات

۴- روش ارزیابی:

بررسی نیازمندی‌های کارکردی سامانه از قبیل:

- سامانه باید توانایی دریافت رویدادهای دسترسی به فایل از سیستم‌عامل را داشته باشد.
- هر فایل دسترسی‌یافته باید با سیاست‌های امنیتی موجود تطبیق داده شود.
- فرآیند تطبیق باید بر اساس ویژگی‌هایی مانند هش فایل، مسیر، نوع دسترسی، و محتوای داخلی انجام‌پذیر باشد.
- سیاست‌های امنیتی باید به‌صورت پویا از پایگاه داده قابل فراخوانی و به‌روزرسانی باشند.
- تصمیم نهایی (مانند اجازه یا مسدودسازی) باید به‌صورت بلادرنگ اعمال شود.
- تمام تصمیمات و رویدادهای امنیتی باید در فایل لاگ ثبت شوند.
- نتایج تحلیل فایل‌ها باید در حافظه کش ذخیره شوند تا از پردازش تکراری جلوگیری شود.
- فایل‌هایی که بارها بررسی شده‌اند یا از کاربران قابل اعتماد آمده‌اند، باید در لیست بهینه‌سازی دسترسی قرار گیرند.
- اطلاعات کش، تحلیل‌ها و سیاست‌ها باید در پایگاه داده به صورت ساختاریافته ذخیره شوند.
- سیاست‌ها باید بتوانند ترکیبی از شرایط مختلف را بررسی کنند (مثلاً هش + محتوا + نوع دسترسی).
- سامانه باید امکان ثبت تغییرات در سیاست‌ها را برای پیگیری تغییرات فراهم کند
- نتایج تصمیم‌گیری باید از طریق داشبورد مدیریتی قابل مشاهده باشند.
- سامانه موظف است در فرآیند تحلیل محتوا، از استخراج متن از فرمت‌های رایج شامل دست‌کم فایل‌های PDF و DOCX پشتیبانی کند.
- تحلیل محتوای فایل باید تنها در صورتی انجام شود که سیاست‌های تعریف‌شده به بررسی محتوایی نیاز داشته باشند.
- در صورت بروز خطا در فرآیند تحلیل یا پردازش (مثلاً عدم توانایی در دیکود فایل یا خطای پایگاه داده)، سامانه باید وضعیت ناموفق را ثبت و دسترسی را محدود یا گزارش کند.
- در تصمیم‌گیری‌های نهایی، امکان اختصاص اکشن‌های چندگانه (مانند هم‌زمان لاگ و بلاک) باید پشتیبانی شود.

بررسی نیازمندی‌های غیرکارکردی سامانه مانند:

- سامانه باید به گونه‌ای طراحی شود که افزودن قوانین جدید بدون نیاز به توقف کل سامانه انجام‌پذیر باشد.
- سامانه باید از بروز خرابی و تاخیر در زمان بار بالا یا ورودی غیرمجاز جلوگیری کند.
- تاخیر تصمیم‌گیری برای هر فایل نباید محسوس باشد.
- سامانه باید توانایی پردازش هم‌زمان چند رویداد دسترسی را بدون افت محسوس عملکرد داشته باشد.
- آمارهای کلیدی سامانه مانند تعداد تصمیم‌های مجاز/ممنوع، درصد برخوردهای کش و حجم لاگ تولیدشده قابل جمع‌آوری و گزارش باشند.
- کد سامانه باید به گونه‌ای ساختاردهی شود که اجزای آن به صورت مستقل قابل به‌روزرسانی و تست باشند.
- سیاست‌های امنیتی ذخیره‌شده در پایگاه داده باید در برابر دستکاری غیرمجاز مقاوم باشند.

۵- مراحل انجام و زمان‌بندی پروژه:

مراحل پیاده‌سازی سامانه به صورت زیر است (شکل ۱,۵):

- تحقیق و مطالعه: در این مرحله به مطالعه در مورد نحوه استفاده از API های Fanotify و تعیین حساسیت پرونده ها بر اساس تنظیمات کاربر انجام خواهد شد..
- انتخاب روش پیاده‌سازی: نیاز است تا روش بهینه برای پیاده سازی انتخاب شده و مورد بررسی قرار بگیرد.
- پیاده‌سازی ابزار: در این مرحله پیاده‌سازی بخش‌های مختلف ابزار صورت می‌گیرد.
- ارزیابی و بهبود: پس از پیاده‌سازی ابزار، نیاز است تا کارکرد آن ارزیابی شده و اشکالات آن برطرف شوند.

مراحل	توضیحات	زمان‌بندی
تحقیق و مطالعه	بررسی و یادگیری مفاهیم مورد نیاز	۵ تا ۶ هفته
انتخاب روش پیاده‌سازی	بررسی روش‌های پیاده‌سازی و انتخاب یکی از این روش‌ها	۴ تا ۵ هفته
پیاده‌سازی ابزار	پیاده‌سازی بخش‌های مختلف ابزار و درکنار یکدیگر قرار دادن آن‌ها	۲ تا ۳ ماه
ارزیابی و بهبود	ارزیابی ابزار و رفع اشکالات احتمالی	۱ تا ۲ ماه

شکل ۱,۵: جدول زمانبندی پروژه

۶- امکانات لازم (ابزارها، محیط‌ها، و نرم‌افزارهای مورد استفاده):

ابزار و محیط مورد نیاز برای پیاده‌سازی سامانه، به صورت زیر است:

- لینوکس (برای استفاده از ابزارهای سطح هسته مانند Fanotify)
- زبان C، C++ یا Rust(برای توسعه و ارتباط با سیستم‌عامل و استفاده از ابزارهای مختلف)
- IDE (برای نوشتن و ویرایش کد)
- GCC (برای کامپایل کردن کدهای C)
- GDB (برای دیباگ کردن کدهای C)

- Git (برای مدیریت نسخه‌های کد)

۷- مراجع و منابع:

[1] Kerrisk, Michael. "The Linux Programming Interface: A Linux and UNIX System Programming Handbook." No Starch Press, 2010.
[2] Smalley, Stephen D., et al. "Security-Enhanced Linux (SELinux) Policy Guide." NSA, 2001.
[3] Friedl, Jeffrey E.F. "Mastering Regular Expressions." O'Reilly Media, 2006.
[4] "Linux Fanotify: A Filesystem Monitoring API" - *Linux Journal*, https://www.linuxjournal.com