TRABAJO DE GRADO
Seminario.

Antivirus de Archivos S3 en la nube con Django y ClamAV

Corporación Universitaria Remington.
Facultad de Ingeniería.
Ingeniería de Sistemas.

Autores:
Johan Ederlíen Luna Bermeo. CC. 1.143.966.442
Efraín Díaz Castillo. CC. 1.193.039.635
Tutor:
Mg. Daniel Fernando Arteaga
Seminario.
2024

Dedicatoria

Con todo el amor y gratitud, dedico este trabajo a mi esposa Fernanda. Gracias por la paciencia, apoyo incondicional y por ser el respaldo durante todo este proceso.

Y a mi familia, que siempre ha creído en mí y me ha motivado a seguir adelante. Este logro es fruto de su amor y apoyo incondicional.

Johan Ederlíen Luna Bermeo.

Con profunda gratitud y amor dedico este logro a ustedes, gracias mamá y papá por su sacrificio y por enseñarme el valor del esfuerzo y la perseverancia. A mis hermanos por ser mi apoyo constante y por creer siempre en mi incluso en los momentos más difíciles. Sin ustedes no sería posible este éxito es tan suyo como mío, con todo mi cariño.

Efraín Díaz Castillo.

Agradecimientos

Se expresa el más sincero agradecimiento al profesor Daniel Fernando Arteaga por su invaluable apoyo y enseñanza práctica durante el proceso de formación del seminario de grado. Su guía ha sido crucial para trazar y enfrentar la realidad que abordaremos como futuros ingenieros en sistemas.

Gracias profesor Arteaga por su paciencia, su tiempo y por compartir su vasta experiencia, la cual se evidencia clase tras clase. Este logro es en gran parte gracias a él.

Tabla de Contenidos

Resumen

Una empresa startup colombiana del mundo de importaciones, de la cual nos reservamos su nombre por temas de confidencialidad, al igual que muchas otras del mercado que tienen su infraestructura en AWS (Amazon Web Services), realiza el almacenamiento de datos confidenciales que soportan en flujo de sus operaciones en buckets de uno de los servicios más populares AWS conocido como Amazon S3. Estos sistemas de almacenamiento están integrados en sus entornos productivos para brindar funcionalidad a su plataforma web. Sin embargo, una limitación significativa de este servicio es la ausencia de un sistema integrado de detección de virus.

Para abordar esta necesidad crítica, se desarrollará un sistema que permita procesar eventos y escanear archivos que finalmente reposarán en Amazon S3 utilizando un motor de antivirus open source, siendo ClamAV, a nuestra consideración, el más adecuado.

Este proyecto no solo proporcionará una capa adicional de seguridad para los datos almacenados en Amazon S3, sino que también demostrará cómo se pueden integrar herramientas de código abierto con servicios en la nube para crear soluciones robustas y escalables. La implementación de este sistema mejorará significativamente la seguridad e integridad de los datos, beneficiando a las organizaciones que operan en entornos de nube.

Palabras clave: Desarrollo de Software, Python, Framework Django, Seguridad en la Nube, ClamAV.

Pregunta orientadora de la búsqueda

El concepto de seguridad en la nube es de vital importancia en la era digital actual y está estrechamente ligado a la protección de los activos más importantes de una organización como lo es su información, focalizando en empresas que almacenan datos sensibles en servicios como Amazon S3. Aunque Amazon Web Services (AWS) ofrece soluciones de almacenamiento robustas, seguros y escalables estos servicios, por sí solos carecen de un sistema integrado para la detección temprana de virus para archivos en tránsito que finalmente reposan en este sistema de almacenamiento.

De acuerdo a un comunicado de prensa digital de Kaspersky (2023), sus sistemas de detección descubrieron un promedio de 411,000 archivos maliciosos diarios en 2023 con un aumento de casi el 3% en comparación con el año 2022. Esto nos alerta sobre el riesgo en portales transaccionales que permiten a los usuarios externos subir ficheros sin una detección adecuada.

A finales del año 2023, la empresa en cuestión asumió la labor de solucionar un hallazgo similar; resultado de una prueba de Ethical Hacking tipo caja negra, lo cual llevó a la necesidad de dicha solución que nos lleva a la pregunta:

¿Cómo diseñar y desarrollar un sistema eficaz para la detección de virus en archivos almacenados en Amazon S3, utilizando micro servicios en Python incorporando el framework Django y el antivirus de código abierto ClamAV, para mejorar la seguridad de los datos en entornos de nube corporativos?

Tabla 1. Indicadores de Ciberseguridad 2023 de Kaspersky

INDICADOR	**ESTADÍSTICA**
Archivos maliciosos detectados diariamente	411,000
Incremento de ataques con documentos maliciosos	53 %
Archivos maliciosos detectados en total en 2023	125 millones
Porcentaje de ataques dirigidos a Windows	88 %
Porcentaje de amenazas diseminadas por scripts y documentos	10 %
Incremento en ataques con archivos PDF de phishing	24,000 archivos
Crecimiento en el uso de puertas traseras	15,000 a 40,000 archivos por día
Archivos maliciosos detectados con puertas traseras en 2023	40,000 archivos por día

Metodología de búsqueda de la información

La metodología de la búsqueda de información se ha estructurado en varias fases, cada una de ellas con actividades específicas para llegar a la solución de la necesidad identificada. A continuación, se describen detalladamente:

Fase 1: Identificación de la necesidad

La empresa en cuestión, de la cual no revelamos información por motivos de confidencialidad, necesita crear un sistema para escanear los archivos PDF que suben algunos clientes externos como parte de su flujo de operación. El criterio de aceptación que permite reutilizarse de manera libre radica en que el procesamiento debe ser preventivo y activarse mediante un evento de almacenamiento, de manera que no imponga carga adicional a la plataforma transaccional.

Fase 2: Conceptualización y arquitectura de la solución

Se aprovechan los conocimientos adquiridos en el seminario “Manejo de Micro servicios con Python y Rest API”, utilizando Django como framework principal debido a sus múltiples prestaciones, funcionalidad y facilidad de desarrollo. Django, combinado con REST API que proporciona una estructura robusta y eficiente para la implementación de micro servicios en Python, adicional se complementa con la solución de antivirus ClamAV para realizar personalización en código.

Fase 3. Búsqueda de Referencias en Internet

Se investigaron arquitecturas similares publicadas en internet aprovechando el sin número de proyectos tecnológicos y repositorios Github públicos que sirven de referencia para utilizarse

como punto de partida y fortificar una solución orientada a satisfacer la necesidad específica de Amazon Web Services con el servicio Amazon S3.

Palabras clave para las búsquedas en internet:

- *"virus detection in Amazon S3"*
- *"AWS S3 file scanning architecture"*
- *"microservices architecture for cloud security"*
- *"ClamAV integration with AWS"*
- *"Python Django AWS integration"*

Los buscadores y bases de datos utilizados fueron Google Scholar, Google Academy, ACM Digital Library, y documentación oficial de AWS.

Sustentación teórica de la pregunta

Título 1. Seguridad en la Nube y Almacenamiento de Datos

Titulo 1.2. Importancia de la seguridad en la nube:

La seguridad en la nube es fundamental en la era digital actual, ya que muchas organizaciones dependen de servicios en la nube para almacenar y gestionar grandes volúmenes de datos sensibles. La protección de estos datos contra accesos no autorizados, pérdida de datos y ataques cibernéticos en especial la infección con virus y/o malware es esencial para mantener la confianza y la integridad de la información.

Figura 1. Servicio Amazon s3 de Amazon Web Services.

Titulo 2. ClamAV como Solución Antivirus

Titulo 2.1. Eficacia de ClamAV en la detección de malware

La eficacia de ClamAV en la detección de malware ha sido validada en numerosos estudios y entornos de producción. Su base de datos de firmas de virus se actualiza regularmente, asegurando que pueda identificar las amenazas más recientes y proteger los datos de manera efectiva.

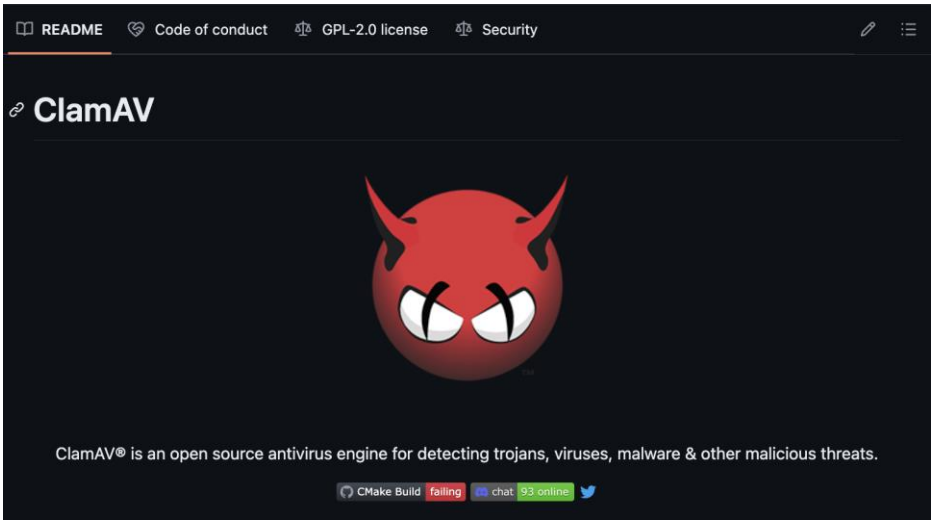

Figura 2. Antivirus Open Source ClamAV - Repositorio Github.

Titulo 3. Arquitectura de Microservicios

Titulo 3.1. Beneficios de los microservicios para la seguridad:

Los microservicios ofrecen varios beneficios en términos de seguridad, incluyendo la capacidad de aislar componentes, implementar controles de seguridad específicos por servicio y escalar de manera independiente. Esta arquitectura facilita la detección y respuesta a incidentes de seguridad, mejorando la resiliencia del sistema.

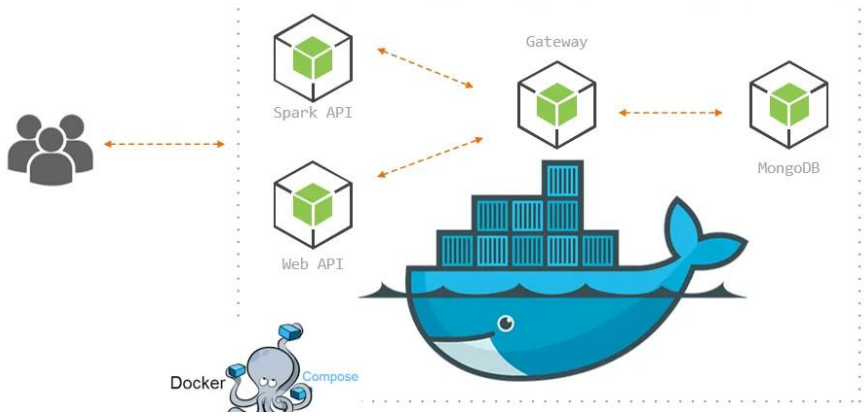

Figura 3. Plataforma de Contenerización Docker para el manejo de microservicios.

Titulo 4. Implementación de Django para Microservicios

Titulo 4.1. Beneficios y características específicas de Django para este proyecto:

Para este proyecto, Django ofrece varias ventajas, incluyendo su facilidad de uso, amplio soporte de la comunidad, y capacidades avanzadas de manejo de datos y seguridad. Estas características permiten una implementación eficiente y segura de los microservicios necesarios para el sistema de detección de virus.

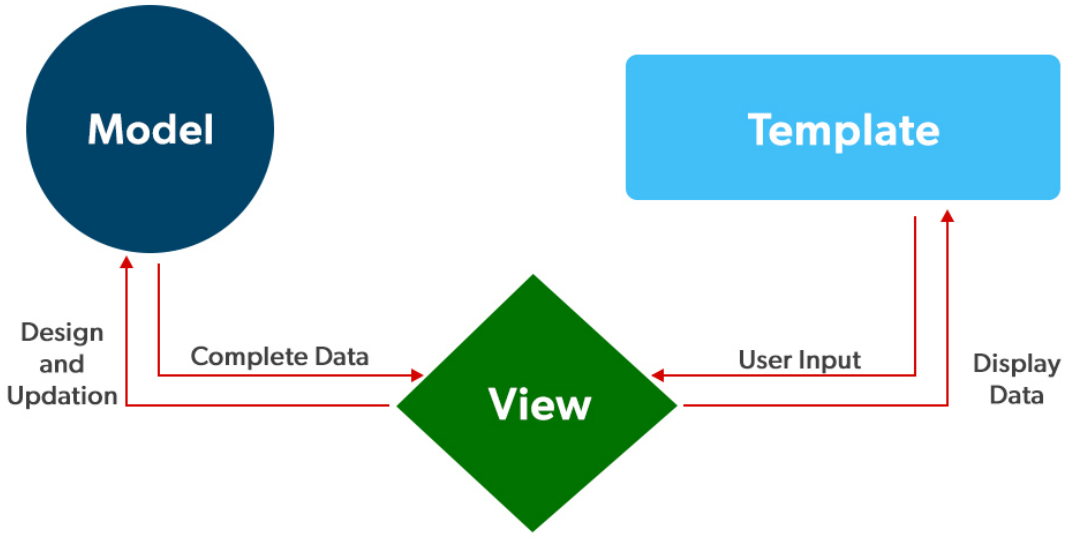

Figura 4. Patrón MTV (Model Template View) de Django Framework

Conclusiones.

- La implementación de un sistema para la detección de virus en archivos almacenados en Amazon S3 utilizando microservicios en Python con Django y ClamAV proporciona una solución robusta y escalable a un problema crítico de seguridad en la nube.
- La revisión teórica de la bibliografía consultada respalda la viabilidad y eficacia de esta solución, destacando la importancia de integrar herramientas de código abierto y arquitecturas modernas para mejorar la seguridad de los datos en entornos de nube corporativos.

Referencias

Amazon Web Services Inc. (2023). Configuring Amazon S3 Event Notifications. Amazon S3. User guide. https://docs.aws.amazon.com/AmazonS3/lastest/userguide/EventNotifications.html.

Khodak, V. (2023). Creating AWS Resources Using Boto3 for Deploying Django Project Vadym Khodak's Blog. https://vadymkhodak.com/blog/posts/creating-aws-resources-using-boto3-for-deploying-django-project.

Amazon Web Services Inc. (2023). Virus Scan S3 Buckets with a Serverless ClamAV-Based CDK Construct. AWS Developer Blog. https://aws.amazon.com/blogs/developer/virus-scan-s3-buckets-with-a-serverless-clamav-based-cdk-construct.

Instil Software. (2023). Scanning Files with ClamAV and CDK. Instil Blog. https://instil.co/blog/scanning-files-with-clamav-and-cdk.

Computer New Age. (2014). Cómo detectar virus en Linux con ClamAV. Computer New Age. https://computernewage.com/2014/10/07/como-detectar-virus-en-linux-con-clamav.

Sutton, T. (2023). Scanning Files with ClamAV on AWS with a NodeJS Fargate SQS Consumer with Terraform. Dev.to. https://dev.to/sutt0n/scanning-files-with-clamav-on-aws-with-a-nodejs-fargate-sqs-consumer-with-terraform-5048.

Amazon Web Services Inc. (2023). Boto3 Configuration. Boto3 Documentation. https://boto3.amazonaws.com/v1/documentation/api/latest/guide/configuration.html.

Toro, G. (2023). Diseño e Implementación de un Escáner de Malware Serverless. RIUNET. https://riunet.upv.es/bitstream/handle/10251/196953/Toro%20-%20Diseno%20e%20Implementacion%20de%20un%20Escaner%20de%20Malware%20Serverless.pdf?

sequence=1&isAllowed=y.

Kaspersky Lab. (2023). Ciberamenazas al alza: 411,000 archivos maliciosos circularon diariamente en 2023. Kaspersky Latin America. https://latam.kaspersky.com/about/press-releases/2023_ciberamenazas-al-alza-411000-archivos-maliciosos-circularon-diariamente-en-2023.

Anexos

Código Fuente

Se anexan los siguientes ficheros de código para explicar la funcionalidad básica del sistema.

Archivo Dockerfile

Permite compilar el código fuente como microservicio para poder desplegar en cualquier orquestador de microservicios como kubernetes.

```
FROM python:3.9-slim
RUN apt-get update && \
    apt-get install -y clamav clamav-daemon && \
    freshclam && \
    apt-get clean
COPY requirements.txt requirements.txt
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
WORKDIR /antivirus_s3/
RUN mkdir -p /tmp/uploads
ENV DJANGO_ENV=development
ENV DJANGO_DEBUG=True
EXPOSE 80
CMD ["python", "manage.py", "runserver", "0.0.0.0:80"]
```

Archivo urls.py

Permite exponer las rutas del api

```
from django.contrib import admin
from django.urls import path
from . import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name='index'),
    path('scan-sqs-messages/',                          views.scan_sqs_messages,
name='scan_sqs_messages'),
    path('scan-messages/',                             views.scan_messages_view,
name='scan_messages_view'),
]
```

Diagramas de Arquitectura

Se anexa el diagrama de arquitectura con todos los componentes transversales que permita entender el flujo inicial para activar los escaneos automáticos de los ficheros.

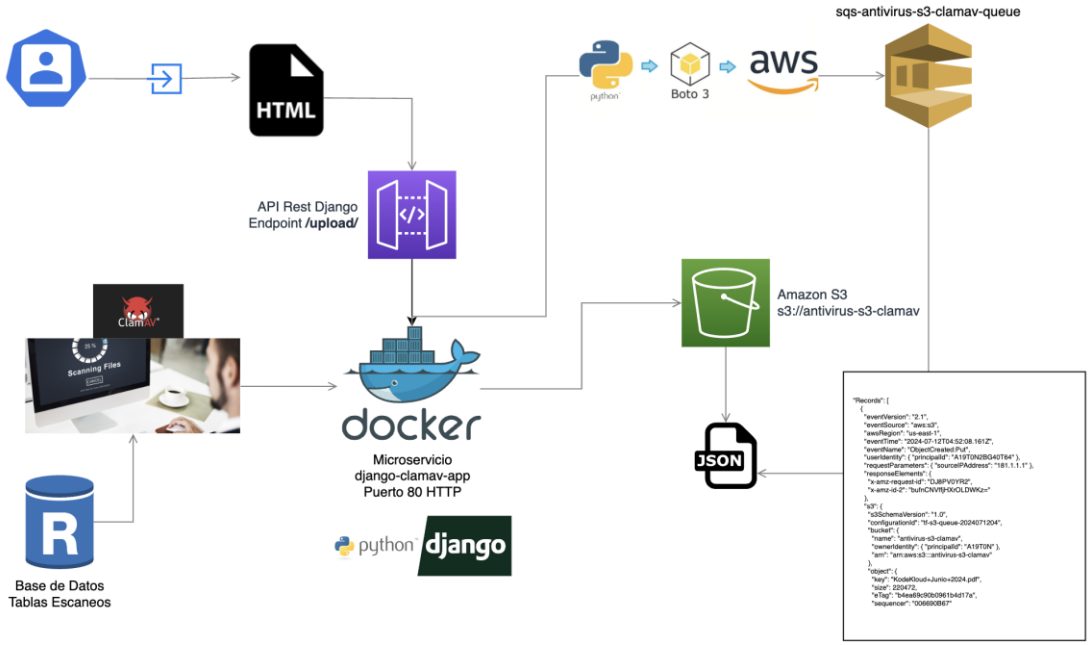

Figura 5. Arquitectura y flujo del sistema.