ARASME: Automatic *Risk assessment* for Small and Medium Enterprises

John García, Mauricio Morales y Sergio Mahecha

Departamento de Ingeniería de Sistemas y Computación, Universidad de los Andes

***Abstract*-** The automation of a *risk assessment* process has not completely been addressed in business systems. Moreover, the small and medium enterprises (SME) struggle to identify and prioritize security risks in their information systems due to lack of resources and expertise. This problem makes any effort of the SMEs to secure and protect their systems against cyberattacks ineffective. To solve this problem, there are several sources of business information that can be used to automatically derive the security risk associated to a given scenario of analysis. This paper describes the development of a software tool intended for automatic *risk assessment* for small and medium enterprises (ARASME). This software tool is based on the FAIR ontology [33] which allows to compute the overall risk as the aggregation of risk factors modeled as probabilistic distributions. In particular, the risks in web applications for SMEs were chosen as the scenario of analysis to demonstrate the functionality of ARASME.

***Index Terms*-** Valoración automática de riesgo, Ontología FAIR, PYME.

I. INTRODUCCION

Aunque existen nuevos enfoques para cuantificar el riesgo de seguridad en los sistemas de información, existen varios retos que necesitan ser solucionados para implementar un proceso automático de valoración de riesgo (*risk assessment*). Los tres (3) principales retos que aborda este trabajo para lograr la automatización del proceso de *risk assessment* son los siguientes: ¿Cómo lograr la agregación de datos heterogéneos provenientes de diferentes fuentes de información? ¿Cómo evitar la dependencia del experto evaluador a la hora de cuantificar el riesgo? ¿Cómo analizar todo el espectro de riesgo que enfrenta una PYME?

El primer reto consiste en poder calcular el valor del riesgo con base en datos con contexto de seguridad proveniente de múltiples fuentes de información, tales como OSINT, *Threat Intelligence*, logs internos, herramientas de escaneo de vulnerabilidades, formularios y encuestas a empleados y, otras más, que suministran datos con una estructura semántica y sintáctica muy diferente. Esta variedad de estructuras hace muy difícil la agregación de información para crear conocimiento acerca del riesgo analizado. Para resolver estas diferencias semánticas y sintácticas, este trabajo usa un tipo de integración basada en ontologías semánticas que permite la agregación de datos provenientes de fuentes heterogéneas de información.

El segundo reto, expone el hecho de que las actuales metodologías de *risk assessment* dependen en gran medida de la experiencia y habilidad del evaluador de seguridad para poder estimar el riesgo. Aún más, algunas de estas metodologías utilizan enfoques cualitativos para calcular el riesgo, lo que hace que los resultados obtenidos sean altamente subjetivos. Este reto es abordado mediante el uso de un enfoque cuantitativo para calcular el riesgo, en lugar del enfoque cualitativo tradicional. La metodología usada en este trabajo es conocida como *Factor Analysis Information Risk* (FAIR) la cual calcula el riesgo con base en distribuciones de probabilidad de ciertos factores de riesgo, tales como la frecuencia de eventos de amenazas (TEF), la capacidad de la amenaza (TCap), la Dificultad (Diff), las Pérdidas de magnitud (LM), y otras adicionales, usando simulaciones de Montecarlo [33].

Finalmente, el último reto que se aborda en este trabajo es seleccionar el tipo de riesgo más significativo para las PYMES. Desafortunadamente, las compañías enfrentan varios de tipos de riesgos de seguridad que van desde los conocidos ataques de ingeniería social, pasando por el riesgo operativo, el de cadena de suministro, hasta incluso los elaborados ataques de inyección de código en aplicaciones web. Realizar un análisis completo de todo este amplio espectro de riesgos representa una labor compleja, demorada y que no puede ser automatizada fácilmente. Por lo tanto, fue necesario delimitar el escenario de riesgo bajo el cual la herramienta de automatización podía ser implementada y evaluada. Es así, que la herramienta ARASME fue construida para analizar el riesgo de seguridad de la información asociado al uso de aplicaciones web en PYMES. Específicamente, el primer prototipo de la herramienta ARASME fue diseñado para automatizar el proceso de estimación de riesgo asociado a ataques de inyección de código SQL, el riesgo asociado a falencias en los protocolos de autenticación y autorización, y al riesgo de explotación de vulnerabilidades no remediadas de aplicaciones web, en empresas con presencia digital en internet. Sin embargo, esta prueba de concepto puede ser extendida a escenarios diferentes sin perder aplicabilidad.

II. CONTEXTO

Resolver el problema de la automatización del proceso de *risk asesssment* requiere primero una comprensión del estado del arte. Por un lado, varios trabajos han estudiado recientemente el problema de automatizar la estimación de riesgo [11,19,20]. Sin embargo, ninguno de ellos ha ofrecido una solución completa al problema. Ciertamente, este campo de estudio requiere mayor

investigación y esfuerzo académico. Por otro lado, estudios recientes han abordado el problema del cálculo de riesgo mediante un enfoque cuantitativo [34,49,50]. La mayoría de estos trabajos están basado en modelos matemáticos que son aplicables únicamente a escenarios particulares de análisis y su alcance es muy limitado.

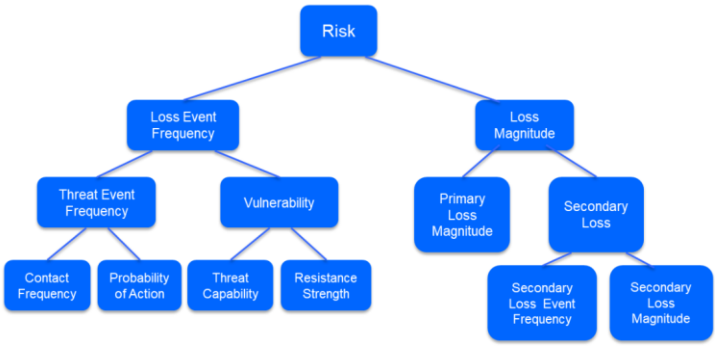
Finalmente, el reto de la agregación de datos provenientes de fuentes de información diversas ha sido estudiado recientemente [48,51,52]. Dos (2) diferentes enfoques fueron revisados: El primero, realizando la integración de los datos mediante procesos ETL (*Extract, transform and load*) utilizados para filtrar, depurar y homologar los datos, con el objetivo final de poder aplicar cierta lógica que produzca un resultado derivado de las múltiples fuentes de información originales [53]; El segundo, construyendo un modelo jerárquico basado en las relaciones de conceptos en un dominio de conocimiento, en donde las relaciones y propiedades de dichos conceptos quedan explicitadas. Este modelo se denomina ontología semántica [48].

III. TRABAJO RELACIONADO

Los marcos de trabajo (*Frameworks*) y las metodologías son esenciales para alcanzar resultados estructurados y consistentes en el proceso de *risk assessment*. De hecho, muchos de ellos existen para ayudar al evaluador con la laboriosa tarea de evaluar el riesgo. Entre los *Frameworks* y metodologías más populares en el ámbito académico y empresarial están: ISO27005, NIST 800-30, MAGERIT, OCTAVE, y FAIR.

A. Metodología de Risk assessment

Este trabajo hace uso de la metodología FAIR (*Factor Analysis of information Risk*) desarrollado por Jack Jones en el año 2006 [33], y que está basado en la ontología de FAIR. En sus orígenes este método cuantificaba el riesgo basado en el enfoque de redes bayesianas. Posteriormente, el método sustituyó el cálculo del riesgo a través de las simulaciones de Montecarlo, debido a la reducción en complejidad y procesamiento que ofrecía este nuevo método. La ontología de FAIR utiliza los factores de riesgo, ilustrados en la siguiente imagen tomada de [56], y son usados para estimar el valor del riesgo:



B. Cálculo del riesgo

Existen diferentes enfoques para estimar el riesgo. Por un lado, un evaluador de seguridad puede clasificar el riesgo usando diferentes categorías para distinguir su prioridad, tales como, "Alto", "medio" y "bajo". Este método de estimación de riesgo es conocido como enfoque cualitativo, y depende en gran medida de la habilidad y experiencia del evaluador, por lo que es altamente subjetivo. Por otro lado, existe un enfoque completamente diferente, donde el riesgo es calculado mediante ecuaciones matemáticas donde los factores de riesgo constituyen las variables de la ecuación. Algunas ecuaciones calculan el riesgo como el producto de los factores de riesgo denominados probabilidad de ocurrencia e impacto del evento. Sin embargo, el riesgo es mucho más complejo que esa simple ecuación y, con el paso del tiempo se ha reformulado para considerar otros elementos en juego [34]. En la metodología FAIR, los factores de riesgo son considerados distribuciones probabilísticas que ayudan a mejorar la veracidad de los resultados, a expensas del grado de exactitud [54]. La idea que hay detrás de este concepto, es reducir el nivel incertidumbre sobre el valor del riesgo en lugar de calcular su valor exacto. Para ello, se emplean las siguientes ecuaciones tomadas de [34], que describen las relaciones entre los factores de riesgo propuesta por la ontología FAIR [55,56]:

Ecuación	Factor de Salida	Factores de Entrada
$L_T = L_P + L_S$	Total Loss (TL): L_T	Primary Total Loss (PTL): L_P
		Secondary Total Loss (STL): L_S
$L_P = RA\ (F_P, LM_P)$	Primary Loss (PL): L_P	Primary Loss Event Frequency (PLEF): F_P
		Primary Loss Magnitude (PLM): LM_P
$L_S = RA(F_S, LM_S)$	Secondary Loss (SL): L_S	Secondary Loss Event Frequency (SLEF): F_S
		Secondary Loss Magnitude (SLM): LM_S
$M_{PLEF} = F_{TE} \times P_V$	Mean of Primary Loss Event Frequency (MPLEF): M_{PLEF}	Threat Event Frequency (TEF): F_{TE}
		Vulnerability (V): P_V
$F_P = Poisson(\lambda = M_{PLEF})$	Primary Loss Event Frequency (PLEF): F_P	Mean of Primary Loss Event Frequency (MPLEF): M_{PLEF}
$F_S = Binomial(n = F_P, p = P_{SL})$	Secondary Loss Event Frequency (SLEF): F_S	Primary Loss Event Frequency (PLEF): F_P
		Chance of Secondary Loss (CSL): P_{SL}
$F_{TE} = F_C \times P_A$	Threat Event Frequency (TEF): F_{TE}	Contact Frequency (CF): F_C
		Probability of Action (PoA): P_A
$P_V = P(P_{TC} > P_{RS})$	Vulnerability (V): P_V	Threat Capability (Tcap): P_{TC}
		Resistance Strength (RS): P_{RS}

RA = Risk Aggregation
P = Probability

Afortunadamente, estos cálculos no deben realizarse de forma manual en la metodología FAIR, y en lugar de ello, se utilizan herramientas que permiten reducir la complejidad de los cálculos. Algunas de estas herramientas son, por ejemplo, modelos matemáticos construidos en Excel [57,58], software comercial (Risklens) [59], software gratuito de entrenamiento (FAIR-U) [60] y librerías de software como Pyfair [61], que utilizan simulaciones de Monte Carlo [33,34] para facilitar el cálculo del riesgo.

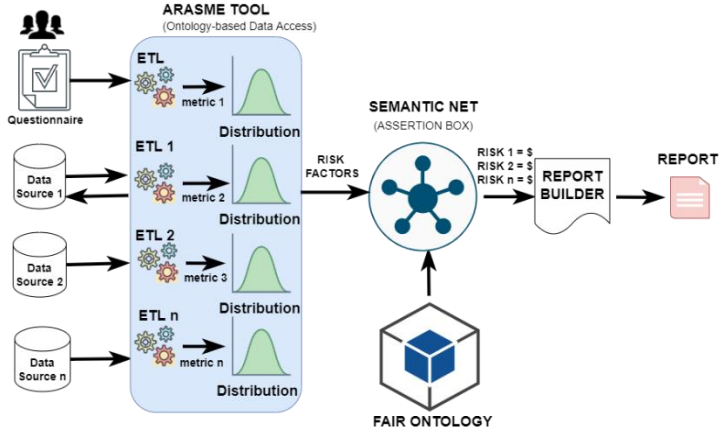
IV. ARASME: LA SOLUCION

Este trabajo propone una herramienta de software denominada ARASME que permite automatizar el cálculo del valor de riesgo haciendo uso de la ontología de FAIR.

Los requerimientos funcionales de esta herramienta son los siguientes:

- Recopilar datos con contexto de seguridad del negocio, provista de forma manual por el usuario, usando un formulario o cuestionarios para este fin.
- Capturar datos de seguridad de la organización de forma automática, por medio de herramientas de adquisición y utilizando fuentes de información de consulta pública e internas de la empresa.
- Adquirir, procesar, refinar y cuantificar estos datos con algoritmos automatizados.
- Alimentar el Algoritmo de la librería Pyfair con los cálculos y estadísticas del estimador.
- Identificar y evaluar los riesgos de mayor criticidad para el negocio.
- Calcular automáticamente, a partir del anterior, el valor del riesgo para las diferentes amenazas analizadas, aplicando ontología de la metodología FAIR.
- Generar de forma automatizada, el respectivo informe de *Risk assessment* que, además de calcular los riesgos de forma cuantitativa, contendrá la priorización de estos.
- Ejecutar el proceso de *Risk assessment* bajo solicitudes específicas de los usuarios (manual) o de forma periódica programada.

La arquitectura de alto nivel de esta herramienta se muestra a continuación:



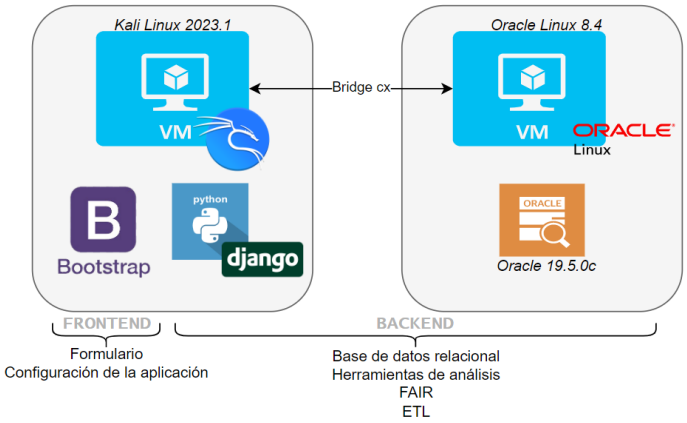
Los principales componentes de la herramienta ARASME son los bloques ETL (*Extract, transform and load*), los módulos de distribución de probabilidad, el modelo de ontología FAIR, la red semántica, y finalmente, el generador de reportes. Los bloques ETL son utilizados para extraer, convertir y almacenar los datos provenientes de fuentes heterogéneas para reducir la carga de procesamiento de las etapas posteriores. Los módulos de distribución de probabilidad son los encargados de transformar las métricas de seguridad en distribuciones de probabilidad asociadas a los factores de riesgo (*Contact frecuency, Probability of action, Threat capability and Difficulty*). La implementación del modelo de la ontología de FAIR contiene las relaciones matemáticas entre los factores de riesgo que son usados para calcular la distribución de probabilidad del riesgo. La red semántica es el módulo encargado de cargar y extraer datos de la ontología, para obtener conocimiento del modelo. Finalmente, el módulo generador de reportes es el encargado de construir el reporte final presentado al usuario.

V. IMPLEMENTACION DE LA HERRAMIENTA

Se han establecido dos máquinas virtuales para analizar riesgos y vulnerabilidades de red. La primera, utilizando Kali Linux, incluye todas las herramientas necesarias para realizar análisis de vulnerabilidades, además de alojar la aplicación ARASME y su repositorio de archivos. La red de las máquinas se configura en modo puente para un fácil escaneo en la red empresarial.

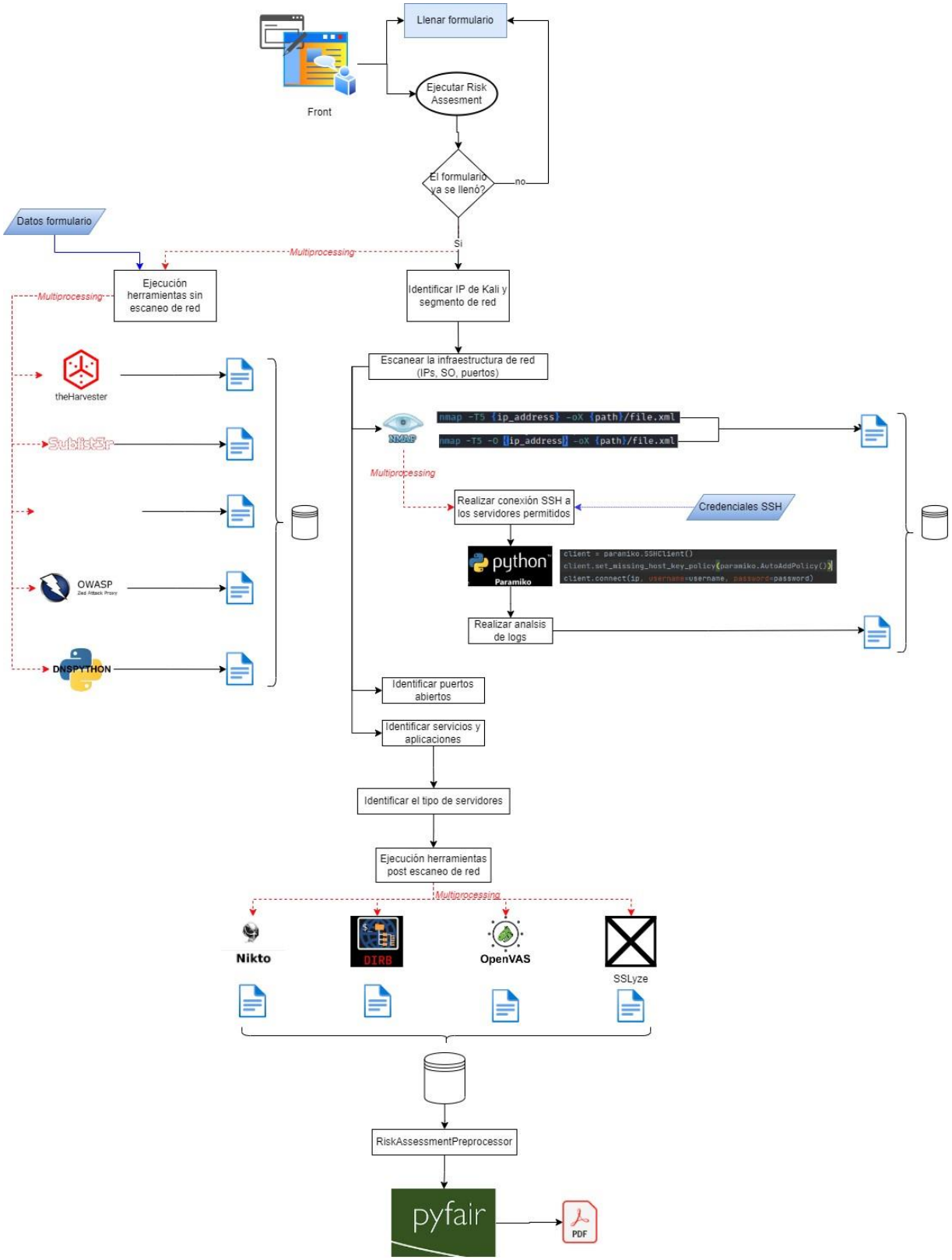
La segunda máquina, basada en Oracle Linux, tiene una base de datos Oracle 19c para almacenar temporalmente información sensible durante la evaluación de riesgos. La información se guarda en tablas temporales y se borra al finalizar el proceso.

En cuanto al *frontend*, se usa Bootstrap para el diseño, incluyendo un formulario seguro para la entrada de datos y configuración de parámetros. El *backend* en Python, modulariza todas las herramientas de Kali, conexiones API, y procesos de transformación de datos para facilitar el incremento de características y la incorporación de nuevas herramientas y metodologías.



VI. FLUJO DE DATOS DE LA APLICACION

El proceso de identificación y análisis de vulnerabilidades, así como la consulta de información en fuentes abiertas (OSINT), demanda un enfoque sistemático y eficaz para asegurar la obtención de resultados de manera óptima. Una ejecución secuencial de tareas se prolongaría excesivamente; por lo tanto, la ejecución en paralelo surge como una alternativa preferible que optimiza el tiempo, la memoria y el uso del procesador. La siguiente imagen muestra el flujo de datos de la herramienta ARASME:


Llenar formulario
Front
Ejecutar Risk Assesment
El formulario ya se llenó?
no
Si
Datos formulario
Multiprocessing
Ejecución herramientas sin escaneo de red
Identificar IP de Kali y segmento de red
Escanear la infraestructura de red (IPs, SO, puertos)
theHarvester
Sublist3r
OWASP
DNSPYTHON
nmap -T5 {ip_address} -oX {path}/file.xml
nmap -T5 -O {ip_address} -oX {path}/file.xml
Realizar conexión SSH a los servidores permitidos
Credenciales SSH
Paramiko
Realizar analsis de logs
Identificar puertos abiertos
Identificar servicios y aplicaciones
Identificar el tipo de servidores
Ejecución herramientas post escaneo de red
Nikto
DIRB
OpenVAS
SSLyze
RiskAssessmentPreprocessor
pyfair
PDF

Antes de iniciar con el proceso de *Risk assessment*, el usuario o cliente debe completar un formulario donde se registra información básica de la empresa, como número de empleados, cantidad de proveedores y preguntas relacionadas con controles y políticas de seguridad, tales como las buenas prácticas y estándares aplicados. Una vez iniciado el proceso de *Risk assessment* la aplicación bifurca el proceso en dos tareas primordiales: la ejecución de las herramientas cuya fuente es el formulario y el escaneo inicial de la red. Estas dos tareas independientes, a su vez activan múltiples herramientas que operan de manera simultánea, cada una focalizada en un aspecto específico de análisis, ya sea el análisis de dominios, búsquedas de correos, análisis de vulnerabilidades en páginas web, identificación de hosts, puertos, servicios, aplicaciones o análisis de vulnerabilidades de servidores web. Este enfoque de ejecución en paralelo es posible gracias a la autonomía e independencia de las tareas. El proceso concluye con la recolección y almacenamiento de la información obtenida en la base de datos de cada una de las herramientas.

La ejecución paralela de estos procesos es facilitada gracias a la librería *Multiprocessing* de Python. Esta librería permite la operación simultánea de múltiples procesos independientes, cada uno con su propio espacio de memoria reservado. Esta práctica difiere de la utilización de hilos o "*Threads*", puesto que cada proceso puede aprovechar su propio núcleo de procesador, en lugar de compartir uno. Esto resulta en un aprovechamiento más eficiente de los recursos del procesador, mejorando el rendimiento durante las fases de escaneo de red y análisis de vulnerabilidades.

Adicionalmente, cada flujo de las herramientas es controlado mediante un manejo adecuado de excepciones, lo que permite que cualquier eventualidad no afecte el funcionamiento normal de la aplicación. Adicionalmente, se preparó la base de datos para tolerar la falta de información de alguna de las fuentes, sin que ello impacte el proceso de análisis final basado en el marco de trabajo FAIR.

Por otro lado, gracias al *Framework* Django, se garantiza la gestión de transacciones, la validación de datos y la realización sencilla de consultas, filtrado y ordenamiento de la información en base de datos, asegurando así la persistencia segura de la información. Finalmente, para el seguimiento de ejecución y la depuración de errores, se están imprimiendo los logs/procesos en la consola con el mayor detalle posible de cada herramienta. Sin embargo, en un futuro se podría considerar la generación de archivos que contengan logs por cada herramienta, permitiendo así una forma más amigable y eficaz de validar los errores desde la interfaz de la aplicación o desde archivos .log.

VII. FUENTES DE INFORMACION

Se identificaron cinco posibles fuentes de información clave para la ejecución del análisis de riesgos; Herramientas de escaneo, las fuentes OSINT (Inteligencia de Fuentes Abiertas), las fuentes internas (registros del sistema operativo), reportes e informes oficiales de amenazas (*Threat Intelligence* y estadísticas) y datos de entrada por medio de formulario en el que el usuario proporciona información específica sobre la organización y el estado actual de esta. Estas fuentes nos permiten combinar información tanto del contexto del negocio como del estado actual de las aplicaciones e infraestructura. Cada una de estas fuentes ofrece un valor único y sustancial, y juntas se complementan de manera ideal para proporcionar un panorama de seguridad completo y confiable. El resultado es un análisis de riesgo que un usuario, propietario o miembro de una organización puede entender en términos monetarios. Este análisis puede ayudar a demostrar las pérdidas financieras significativas debido a la materialización de eventos de seguridad, que es el objetivo principal de nuestra aplicación.

Por lo tanto, estas fuentes no solo son una parte integral de nuestra estrategia de evaluación de riesgos, sino que también forman una combinación exclusiva y potente para recopilar y analizar la información necesaria para comprender y gestionar los riesgos de seguridad de una manera eficaz y fácilmente comprensible.

1) Herramientas de extracción de datos:

Para la selección de las herramientas de extracción de información de seguridad, que fueron usadas en la implementación de ARASME, y que hacen parte de la primera etapa del módulo ETL (*Extract, Transform and Load*), se analizaron cerca de un centenar de herramientas de software comerciales, *opensource*, *freeware* y scripts, categorizadas con base en la fuente de información de la cual extraen los datos: OSINT, eventos internos de sistema operativo, *Threat Intelligence* y escaneo de vulnerabilidades. De este conjunto de herramientas, se seleccionaron aquellas más relevantes, teniendo en cuenta los siguientes criterios:

- ✓ Usabilidad
- ✓ Facilidad de automatización
- ✓ Compatibilidad con Python
- ✓ Disponibilidad de las herramientas de forma gratuita

Con base en los anteriores criterios, las herramientas seleccionadas en la implementación de ARASME fueron las siguientes:

Categoría	Nombre	Objetivo de la herramienta	Métricas	Factor de riesgo asociado
OSINT	TheHarvester	Obtener cuentas de usuario de la organización expuestas en redes sociales	#Cuentas expuestas públicamente	Probability of Action (PoA)
			#IPs públicas	Contact Frequency (CF)
OSINT	SUblist3r	Enumeración de subdominios públicos de la organización	#Subdominios públicos	Contact Frequency (CF)
ESCANER	BeautifulSoup	Web scraping	#Inputs HTML tags	Contact Frequency (CF)
OSINT	DNSPython	Enumeración DNS	Registros DNS	Contact Frequency (CF)
ESCANER	Nikto	Escaneo de vulnerabilidades web	#Vulnerabilidades	Contact Frequency (CF)
ESCANER	Dirb	Enumeración de directorios y archivos ocultos del servidor web	#Directorios del servidor web	Contact Frequency (CF)
ESCANER	SSLyze	Escaneo de protocolos de seguridad SSL/TLS	#Vulnerabilidades y debilidades	Contact Frequency (CF)
ESCANER	OWASP-ZAP	Escaneo de vulnerabilidades web	#Vulnerabilidades	Contact Frequency (CF)
ESCANER	NMAP	Escaneo de vulnerabilidades	#Host #Puertos abiertos	Contact Frequency (CF)
ESCANER	OpenVAS	Escaneo de vulnerabilidades	Puntaje de severidad	Contact Frequency (CF)

A continuación, se detalla la funcionalidad de cada una de estas herramientas de extracción de datos:

Nmap: Esta será nuestra herramienta principal para escanear hosts, puertos y aplicaciones. Nmap es reconocido por su eficacia y precisión en la exploración de redes.

TheHarvester: Usaremos esta herramienta OSINT para buscar subdominios, direcciones IP y correos electrónicos. TheHarvester es conocida por su habilidad para recopilar grandes cantidades de información en poco tiempo.

Sublist3r: Esta herramienta nos ayudará a listar subdominios de páginas web. Su eficiencia y efectividad la hacen ideal para este propósito.
BeautifulSoup: Este será nuestro web scrapper, utilizado para extraer información de las páginas web. BeautifulSoup es altamente compatible con Python y fácil de usar.

OWASP ZAP Proxy: Utilizaremos esta herramienta como nuestro escáner de vulnerabilidades de aplicaciones web. Este es uno de los proyectos más populares de la comunidad OWASP y es confiable para detectar fallos de seguridad.

DNSPython: Esta herramienta será utilizada para consultas de información DNS. DNSPython ofrece una gran cantidad de funcionalidades y es compatible con Python.

Paramiko: Esta es una biblioteca de Python que permite la conexión SSH a servidores. Nos ayudará en las tareas que requieran extracción de información de otros equipos.

Nikto: Este escáner de servidores web nos permitirá examinar los servidores web en busca de problemas de seguridad y vulnerabilidades.

DirB: Usaremos DirB como nuestro escáner de contenido web. Es eficaz para descubrir directorios y archivos en cualquier servidor web.

OpenVAS (GVM): Este escáner de vulnerabilidades será de gran utilidad para analizar y detectar posibles fallos en la seguridad web.

SSLyze: Esta será nuestra herramienta para el análisis de seguridad de protocolos SSL y TLS en servidores web. SSLyze es un potente analizador de configuraciones SSL/TLS.

2) Datos de entrada mediante formulario:

Para entender el contexto actual de la organización, su postura de seguridad y su información básica, hemos seleccionado una serie de preguntas. Aunque hemos intentado redactar estas preguntas de la manera más sencilla y amigable posible, siempre recomendamos que el equipo técnico de la organización esté presente en el momento de completar el formulario. Este equipo debería tener información de primera mano, lo cual es crucial para obtener datos precisos y confiables. Esta información de calidad es esencial para que el análisis de riesgos genere resultados confiables y representativos de la realidad.

El propósito de estas preguntas es obtener información básica para aplicar la metodología FAIR a los escenarios de riesgo, como la inyección SQL, el acceso no autorizado a la información y la explotación de vulnerabilidades no parchadas. Por lo tanto, las preguntas están enfocadas en temas como las aplicaciones web, los controles de acceso, el cifrado de información, y la aplicación de parches a las aplicaciones y sistemas operativos.

A continuación, se ilustran algunos ejemplos de las preguntas formuladas, categorizadas según el tipo de riesgo del que se desea obtener información:

a) Preguntas básicas:

1. ¿Qué tipo de datos confidenciales almacena la organización? Por favor, seleccione todas las opciones aplicables:
- Datos de identificación personal (PII)
- Registros médicos o de salud
- Datos de empleados
- Secretos comerciales o propiedad intelectual
- Datos legales o judiciales

2. Número de empleados:
Por favor, selecciona una opción:
- Microempresa (1-10 empleados)
- Pequeña empresa (11-50 empleados)
- Mediana empresa (51-100 empleados)
- Mediana empresa (101-150 empleados)
- Mediana empresa (151-250 empleados)

3. Por favor ingrese el dominio de la empresa (.com)

b) Preguntas sobre riesgo de Inyección SQL:

1. ¿Utiliza cifrado para proteger la información de identificación personal (PII), los datos de tarjetas de crédito, los datos de salud o cualquier otro tipo de datos sensibles?
- Sí
- No

2. En una escala del 1 al 10 (si aplica), donde 1 indica un bajo nivel de madurez en seguridad y 10 indica un alto nivel de madurez en seguridad, ¿cómo calificaría la madurez en seguridad de los desarrolladores de software en su organización? Por favor, selecciona una opción:
- 1-2: Los desarrolladores tienen poco o ningún conocimiento de las prácticas de seguridad.
- 3-4: Los desarrolladores tienen algún conocimiento de las prácticas de seguridad, pero a menudo no las implementan.
- 5-6: Los desarrolladores generalmente implementan prácticas de seguridad, pero hay algunas áreas de mejora.
- 7-8: Los desarrolladores tienen un buen conocimiento y aplicación de las prácticas de seguridad.
- 9-10: Los desarrolladores tienen un excelente conocimiento y aplicación de las prácticas de seguridad y están constantemente buscando maneras de mejorar.

c) Preguntas sobre riesgo de acceso no autorizado a la información:

1. ¿Qué porcentaje de sus datos en tránsito (transmitidos a través de redes) está protegido mediante cifrado?
- 0-25%
- 26-50%
- 51-75%
- 76-100%

2. ¿Qué porcentaje de sus datos almacenados (en reposo) está protegido mediante cifrado?
- 0-25%
- 26-50%
- 51-75%
- 76-100%

3. ¿Utiliza cifrado para proteger la información de identificación personal (PII), los datos de tarjetas de crédito, los datos de salud o cualquier otro tipo de datos sensibles?
- Sí
- No

d) Preguntas sobre riesgo de explotación de una vulnerabilidad no parcheada:

1. ¿Con qué frecuencia se aplica el proceso de parcheo en su infraestructura de TI?
- Diariamente
- Semanalmente
- Mensualmente
- Anualmente
- No se realiza de manera regular

2. En una escala del 1 al 10, donde 1 indica que se realiza un escaneo de vulnerabilidades muy raramente y 10 indica que se realiza de forma muy frecuente, ¿con qué frecuencia realiza su organización un escaneo de vulnerabilidades?

e) Preguntas financieras y de productividad

1. ¿Cuántos clientes regulares o habituales tiene la empresa en un período mensual?
2. ¿Cuál es el valor promedio (en COP) de un cliente para su empresa?
3. ¿Cuál es el monto de las ganancias netas de hace uno, dos y tres años?

3) Informes de amenazas y estadísticas de seguridad:

Adicionalmente, como datos de entrada fueron utilizados informes y reportes de seguridad de fuentes oficiales y confiables. Por ejemplo, se incorporaron datos de reportes oficiales del OWASP TOP10 respecto a ataques de inyección SQL, ataques diarios a aplicaciones web durante 2022, y el porcentaje de ciberdelincuentes con motivaciones financieras para realizar ataques. La finalidad de estos reportes es mantener la base de datos actualizada con la información de seguridad más reciente. De esta manera, existe la posibilidad de que periódicamente se integre a nuestra aplicación las últimas cifras y nuevos reportes, garantizando que nuestra evaluación se basa en datos de confianza actualizados.

VIII. CRITERIOS DE SELECCION

La selección de la ontología para la integración de fuentes heterogéneas, la metodología de *risk assessment*, el algoritmo calculador de riesgo y los escenarios de riesgo, fue realizada con base en los siguientes criterios de análisis:

1) Selección de la ontología

Para la selección de la ontología, que permite la integración de fuentes heterogéneas de información, fueron revisadas cerca de 30 ontologías disponibles al público académico, entre ellas Unified Cyber Ontology (UCO) [27,28,32,64], Security Asset Vulnerability Ontology (SAVO) [28], The Security Algorithm-Standard Ontology (SASO) [28], The Cyber Effects Simulation Ontology (CESO) [28], STIX Ontology [64,66], Conceptual Cybersecurity Vulnerability Ontology (CVO) [65], Cyber-investigation Analysis Standard Expression (CASE) [68], RAMSS *Risk assessment* Ontology (Reliability, availability, maintainability, safety, and security) [67], y FAIR [33,34,62], entre otras. Posteriormente, se determinó, mediante los siguientes criterios, la que mejor se ajustaba a la implementación de la herramienta ARASME:

- Asociada al dominio de conocimiento del riesgo de seguridad
- Acogida del público académico y empresarial
- Capacidad de estimar riesgo de seguridad
- Descripción de la ontología disponible
- Facilidad de implementación

Después de calificar cada una de las anteriores ontologías, con base en los anteriores criterios, el mayor puntaje fue obtenido por la ontología FAIR y, por lo tanto, fue seleccionada para poder integrar las diferentes fuentes de información heterogéneas que suministraran datos con contexto de riesgo de seguridad.

2) Selección del calculador de riesgo

Para calcular el valor del riesgo fueron evaluados tres (3) diferentes enfoques, utilizados ampliamente en la academia para cuantificar el riesgo de seguridad. El primero de ellos, son los grafos de ataques, que desarrolla un modelo enfocado en encontrar los posibles puntos de ataque que tiene un sistema de información, y que pueden ser utilizados por un atacante para lograr acceder al sistema [62,63]. El segundo de ellos, son las redes bayesianas, que permiten cuantificar el riesgo mediante la estimación de las distribuciones de probabilidad de ocurrencia de diferentes factores de riesgo, que en conjunto ayudar a reducir la incertidumbre del nivel de riesgo general [34,35,36]. El tercero de ellos es el *Framework* de gestión de riesgo conocido como FAIR (Factor Analysis of information Risk) desarrollado por Jack Jones en el año 2006 [33], y que está basado en la ontología de FAIR. En sus orígenes este método cuantificaba el riesgo basado en el enfoque de redes bayesianas. Posteriormente, el método sustituyó el cálculo del riesgo a través de las simulaciones de Montecarlo, debido a la reducción en complejidad y procesamiento que ofrecía este nuevo método.

Los siguientes criterios fueron utilizados para seleccionar el enfoque más apropiado de cuantificación de riesgo de seguridad:

- Enfoque cuantitativo preferido frente al cualitativo
- Cobertura de los riesgos de seguridad analizados
- Facilidad de implementación
- Facilidad de automatización
- Flexibilidad en la selección de los factores de riesgo utilizados

Con base en los anteriores criterios de selección, los métodos de estimación de riesgo basados en redes bayesianas y FAIR, tuvieron los puntajes más altos, siendo el último de estos el que obtuvo el mayor puntaje. Adicionalmente, el método FAIR utiliza el concepto de redes bayesianas por dos (2) razones básicas: Primero, en esencia, FAIR descompone el cálculo de riesgo en elementos interrelacionados denominados factores de riesgo [33], de forma similar a como lo hace una red bayesiana. En las primeras versiones de FAIR el cálculo de riesgo se hacía a través de redes bayesianas [33], pero posteriormente simplificó la complejidad de cálculo usando las simulaciones de Montecarlo; Y segundo, el concepto subyacente en las redes bayesianas es la capacidad de inferir conocimiento partir de estimaciones probabilísticas iniciales de un experto y, posteriormente, refinar ese conocimiento gracias a la disponibilidad de nuevos datos [33,37]. De forma similar, el proceso de *risk assessment* en FAIR consiste en reducir el nivel de incertidumbre del riesgo, partiendo de una estimación probabilística a priori y posteriormente refinándose mediante nuevos datos concluyentes. Por las anteriores razones, el método de FAIR fue seleccionado para la cuantificación de riesgo en la implementación de la herramienta de software ARASME, utilizando las ecuaciones que relacionan los factores de riesgo [55,56].

3) Selección de la metodología de *risk assessment*

Fueron analizados varias metodologías y marcos de trabajo (*Frameworks*) para la realización del proceso de *risk assessment*, entre ellas, Octave (Operationally Critical Threat, Asset and Vulnerability Evaluation), Octave S, Octave Allegro, Magerit (Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información), OSSTMM (Open Source Security Testing Methodology Manual) y FAIR (Factor Analysis of Information Risk).

Los criterios utilizados para la selección de la metodología de *Risk assessment* utilizada en la implementación de la herramienta ARASME son los siguientes:

- La metodología es adecuada para las PYMES
- Facilidad de implementación
- El tipo de riesgo es cubierto por la metodología
- Enfoque cuantitativo
- Requiere poco personal para su ejecución
- Flexibilidad en la selección de los factores de riesgo
- Permite usar varias fuentes de información
- Rapidez de implementación

El mayor puntaje obtenido fue el de la metodología FAIR gracias a su facilidad de implementación, rapidez, la flexibilidad en la disponibilidad de fuentes de información diversas, y la capacidad de realizar un cálculo cuantitativo del riesgo. Por esta razón, la implementación de la herramienta ARASME está basada en la metodología FAIR para la ejecución del *risk assessment*.

4) Selección de los escenarios de riesgo a evaluar

A pesar de que las empresas de pequeña y mediana escala experimentan en sus organizaciones varios tipos de riesgo de seguridad, algunos de ellos son los más representativos, y causan un mayor impacto dentro de este tipo de negocios. Para poder determinar un escenario de análisis representativo del entorno real de las PYMES, fueron analizados varios reportes estadísticos de ataques de seguridad en los últimos años en empresas de pequeña y mediana escala [10,24,38,39,40,41,42], encontrando los siguientes escenarios como los más relevantes:

- Ataques de Ingeniería social
- Infección por software malicioso
- Ataques de inyección de código y XSS en aplicaciones web
- Suplantación de identidad
- Fraudes en pagos electrónicos
- Ataque de denegación de servicio
- Robo de activos de información
- Controles de acceso inadecuados
- Ausencia de cifrado
- Múltiples repositorios de información no seguros
- Uso de configuraciones no seguras o por defecto
- Software legado
- Ausencia de plan de remediación de vulnerabilidades
- Cryptojacking

Debido a que cada uno de los anteriores escenarios plantea diferentes tipos de riesgo, los siguientes criterios fueron utilizados para determinar los escenarios de riesgo a evaluar con la herramienta ARASME:

- Criticidad reportada por OWASP TOP-10
- Multiplicidad de fuentes de información para cuantificar su nivel de riesgo
- Número de herramientas de adquisición de información asociada al riesgo
- Factibilidad de análisis cuantitativo

Los puntajes más altos obtenidos en los escenarios de riesgo son los siguientes:

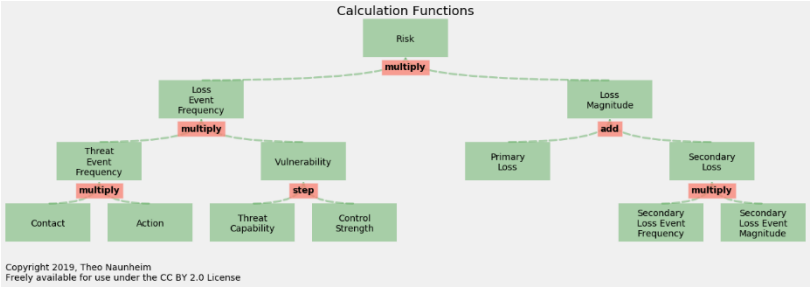
1. Protocolos inseguros de autenticación y autorización (Hacker malicioso)
2. Ataques de inyección SQL (Hacker Malicioso)

Con esto en mente, los anteriores dos (2) escenarios de análisis de riesgo fueron seleccionados, durante la implementación de la herramienta ARASME, para evaluar el riesgo de seguridad de la

información asociado a las aplicaciones web y los activos de información, que soportan las operaciones de las empresas PYME, del sector de los servicios con presencia digital en internet. Es importante aclarar que los anteriores escenarios de análisis fueron seleccionados para reducir la complejidad de análisis e implementación de la herramienta software, y centrarse en demostrar la factibilidad de implementación, pero esta selección no le resta generalidad a la funcionalidad que puede brindar la herramienta ARASME, que podría ser aplicada a otros escenarios de riesgo diferentes.

IX. CUANTIFICACION DEL RIESGO

Para cuantificar el riesgo se utilizan los tipos y rangos de valores de entrada que la librería PyFair define para cada uno de los factores, los cuales son denominados nodos, y que se muestran en la imagen a continuación, tomada de [61].



En PyFair, estos nodos pueden ser proporcionados con cierta flexibilidad. No es obligatorio proporcionar los 7 nodos inferiores para calcular el riesgo. Por ejemplo, se podría facilitar la información relativa al contact, action y vulnerability (omitiendo el threat capability y el control strength), el primary loss y el secondary loss (excluyendo el secondary loss event frequency y el secondary loss event magnitude). Incluso, sería posible proporcionar solo el loss event frequency y el loss magnitude. Lo fundamental es proporcionar los datos tanto de la parte izquierda como de la derecha del árbol para que el proceso se realice sin errores. En el caso de este piloto, hemos decidido proporcionar todos los nodos inferiores, con la excepción de aquellos relacionados con la pérdida secundaria. Para estos, estamos proporcionando directamente la secondary loss, sin incluir el frequency ni el magnitude de las perdidas secundarias.

De acuerdo con la documentación de FAIR (https://pyfair.readthedocs.io/en/latest/), estos son los valores que se espera que reciba cada nodo:

1. Contact Frequency (C)
Descripción: Un vector con elementos que representan la cantidad de contactos de actores de amenazas que podrían generar una amenaza dentro de un período de tiempo determinado.
Entrada: Todos los elementos deben ser un número positivo.

2. Probability of Action (A)
Descripción: Un vector con elementos que representan la probabilidad de que un actor de amenazas proceda después de entrar en contacto con una organización.
Entrada: Todos los elementos deben ser números de 0.0 a 1.0.

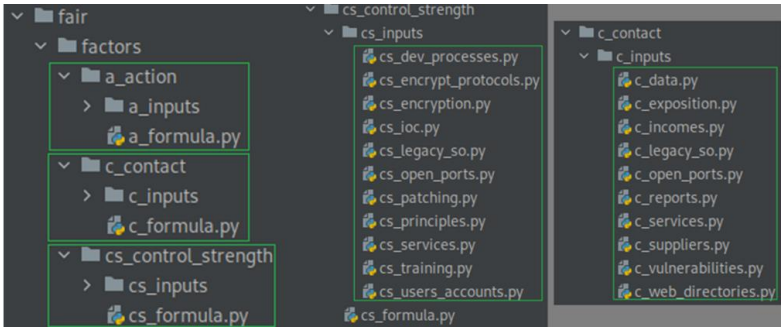
3. Threat Capability (TC)
Descripción: Un vector de elementos sin unidades que describen el nivel relativo de experiencia y recursos de un actor de amenazas (en relación con una Fuerza de control).
Entrada: Todos los elementos deben ser números de 0.0 a 1.0.

4. Control Strength (CS):
Descripción: Un vector de elementos sin unidades que describen la fuerza relativa de un control determinado (en relación con la Capacidad de amenaza de un actor determinado).
Entrada: Todos los elementos deben ser números de 0.0 a 1.0.

5. Primary Loss (PL)
Descripción: Un vector de pérdidas financieras directamente atribuibles a la amenaza.
Entrada: Todos los elementos deben ser un número positivo.

6. Secondary Loss (SL):
Descripción: Un vector de pérdidas financieras atribuibles a factores secundarios.
Entrada: Todos los elementos deben ser un número positivo.

Con la definición de las entradas esperadas por la librería PyFair y las fuentes de información previamente establecidas, hemos desarrollado una lógica para cuantificar el valor de cada uno de estos nodos. Esta lógica facilita la adición modular de fuentes de información y es fácil de implementar a nivel de código. Cada factor de riesgo cuenta con sus respectivas entradas y una fórmula para calcularlo [55,56]. Dentro de cada una de estas entradas, ya sea mediante la extracción o el cálculo de información específica, hemos logrado obtener datos relevantes a través del formulario, el escaneo y los reportes según el riesgo que se está analizando, como se muestra en la siguiente imagen:



Por ejemplo, la cuantificación del riesgo basada en las preguntas que se realizan en el formulario se lleva a cabo de la siguiente manera: En el caso de preguntas de selección múltiple (como la de la imagen a continuación), utilizadas para calcular la frecuencia de contacto, la teoría sostiene que cuantas más opciones se seleccionen, mayor será la frecuencia de contacto.

¿Qué tipo de datos confidenciales almacena la organización? Por favor, seleccione todas las opciones aplicables:

- ☐ Datos de identificación personal (PII) ?
- ☐ Datos financieros
- ☐ Registros médicos o de salud
- ☐ Datos de empleados
- ☐ Secretos comerciales o propiedad intelectual
- ☐ Datos legales o judiciales

Por lo tanto, hemos asignado pesos específicos a cada opción, de acuerdo con su nivel de criticidad, como se ilustra en la siguiente imagen.

```
# Se definen los pesos para
weights = {
    'PII': 15,
    'financial_data': 30,
    'health_records': 15,
    'employee_data': 15,
    'trade_secrets': 10,
    'legal_data': 15,
}
```

Finalmente, se suman todas las opciones seleccionadas.

Por otro lado, en el caso de las preguntas de selección única (como la de la imagen a continuación), también se asigna un valor específico según la criticidad de la opción seleccionada.

¿Con qué frecuencia se aplica el proceso de parcheo en su infraestructura de TI?

Selecciona una opción

- Diariamente
- Semanalmente
- Mensualmente
- Anualmente
- No se realiza de manera regular

Este rango de valores se muestra como referencia en la siguiente imagen:

```
patch_frequency_scores = {
    "daily": 1.0,
    "weekly": 0.8,
    "monthly": 0.6,
    "annually": 0.4,
    "no_regular": 0.0,
}
```

Este valor es una fuente para determinar el control strength y se le asigna un valor a las posibles respuestas que puede variar entre 0.0 y 1.0.

Finalmente, tenemos otro tipo de pregunta de selección única, pero con menos opciones de respuesta (ver imagen abajo).

¿Tiene su organización una política formalizada para la creación de contraseñas que especifica requisitos como longitud mínima, inclusión de números, símbolos, y una combinación de letras mayúsculas y minúsculas?

Selecciona una opción

- Sí, tenemos una política formal y específica
- Tenemos algunas directrices, pero no una política formal
- No, no tenemos una política específica

Al igual que antes, se le asigna un valor que oscila entre 0.0 y 1.0, lo que sirve como otra fuente para determinar el control strength, como se muestra en la siguiente imagen:

```
password_policy_scores = {
    "yes": 1.0,
    "some": 0.5,
    "no": 0.0,
}
```

Para manejar los informes y reportes, hemos creado una tabla en base de datos que almacena toda la información relevante de estos documentos. Esta tabla se compone de varias columnas, como el tipo de factor al que sirve como fuente de información, la descripción del informe, el valor del dato, la medida del valor (por ejemplo, porcentaje, número, cantidad, promedio, etc.) y, finalmente, la URL de la fuente, como se muestra a continuación en la imagen:

	ID	TYPE	DESCRIPTION	VALUE	MEASURE_VALUE	SOURCE
8	8	ACTION	Ataques que usan como vector el robo de credenciales (Año 2022)	14	%	Mandiant. (2023). Special R
9	9	ACTION	Ataques que usan como vector la fuerza bruta (Año 2022)	4	%	Mandiant. (2023). Special R
10	10	ACTION	Aplicaciones web vulnerables a fallos de autenticación en OWASP TOP-10	28	%	OWASP TOP-10 Report(2021)
11	11	CONTACT	MINIMOS ataques diarios a aplicaciones web en el año 2022	50000000	#	Akamai. (2019). Slipping Th
12	12	CONTACT	MAXIMOS ataques diarios a aplicaciones web en el año 2022	160000000	#	Akamai. (2019). Slipping Th
13	13	CONTACT	Ataques a las empresas e-commerce en el año 2022	260000	#	Akamai. (2019). Slipping Th
14	14	CONTACT	Ataques de SQL inyección en el año 2022	5000000000000	#	Akamai. (2019). Slipping Th
15	15	CONTACT	Impacto producido medido mediante el CVSS v3 INJECCION SQL	7,15	avg_scale	OWASP TOP-10 Report(2021)
16	16	CONTACT	Impacto producido medido mediante el CVSS v3 AUTH	5,93	avg_scale	OWASP TOP-10 Report(2021)
17	17	CONTACT	Impacto producido medido mediante el CVSS v3 PARCHES	5	avg_scale	OWASP TOP-10 Report(2021)
18	18	TCAP	Impacto producido medido mediante OWASP INJECCION SQL	7,25	avg_scale	OWASP TOP-10 Report(2021)
19	19	TCAP	Impacto producido medido mediante OWASP AUTH	6,92	avg_scale	OWASP TOP-10 Report(2021)

Este procedimiento se realiza con el objetivo de centralizar y normalizar los valores de estos informes, facilitando su actualización. Por ejemplo, si llega un nuevo informe del año en curso, se puede actualizar fácilmente. Asimismo, facilita la inclusión de nuevos reportes, dado que a nivel de código se filtra por tipo de nodo y se obtienen todos los datos correspondientes. De esta manera, no es necesario realizar modificaciones en el código fuente para agregar un nuevo reporte.

Finalmente, en cuanto a los resultados de las vulnerabilidades, el proceso es un poco más complejo, ya que no es posible estandarizar la respuesta que cada herramienta proporciona, dado que cada una genera resultados de manera única. Por lo tanto, hemos cuantificado la información de manera independiente y personalizada para cada herramienta. Por ejemplo, en el caso de OpenVAS, esta herramienta no genera valores numéricos en los resultados encontrados, sino que simplemente categoriza las vulnerabilidades como bajas, medias o altas (ver imagen abajo).

```python
def get_zap_scores():
    queryset = ZapVulnerabilities.objects.values('risk').annotate(Count('id'))

    zap_score = 0
    for item in queryset:
        risk_level = item['risk']
        count = item['id__count']

        if risk_level in RISK_WEIGHTS:
            zap_score += RISK_WEIGHTS[risk_level] * count
    #print("[FAIR][Contact] Contacto ZAP:", zap_score)
    return zap_score
```

En respuesta a esto, hemos asignado un peso a cada categoría y simplemente sumamos el total de las vulnerabilidades encontradas (ver imagen abajo a modo ilustrativo).

```python
RISK_WEIGHTS = {
    "Low": 1,
    "Medium": 2,
    "High": 3,
}
```

En el caso de Nikto, tampoco genera resultados numéricos en base a los hallazgos (ver imagen a continuación).

```python
def get_nikto_vulns():
    nikto_score = NiktoVulnerabilities.objects.all().count()
    #print("[FAIR][Contact] Contacto Nikto:", nikto_score)
    return nikto_score
```

A pesar de que proporciona información comprensible y fácil de entender en la descripción de los resultados, para este piloto hemos contado directamente los hallazgos y tomado esa cantidad como el resultado final.

Finalmente, a diferencia de los casos anteriores, OpenVAS (GVM) sí genera puntajes según la severidad de los hallazgos, lo que facilita el cálculo final del resultado de esta herramienta (ver imagen abajo).

```python
def get_openvas_scores():
    queryset = OpenvasNVT.objects.exclude(nvt_severities_score='0.0')
    avg_openvas = sum(float(item.nvt_severities_score) for item in queryset)
    avg_openvas = avg_openvas / queryset.count() if queryset.count() > 0 else 0
    avg_openvas = round(avg_openvas)
    #print("[FAIR][Contact] Contacto OpenVas:", avg_openvas)

    return avg_openvas
```

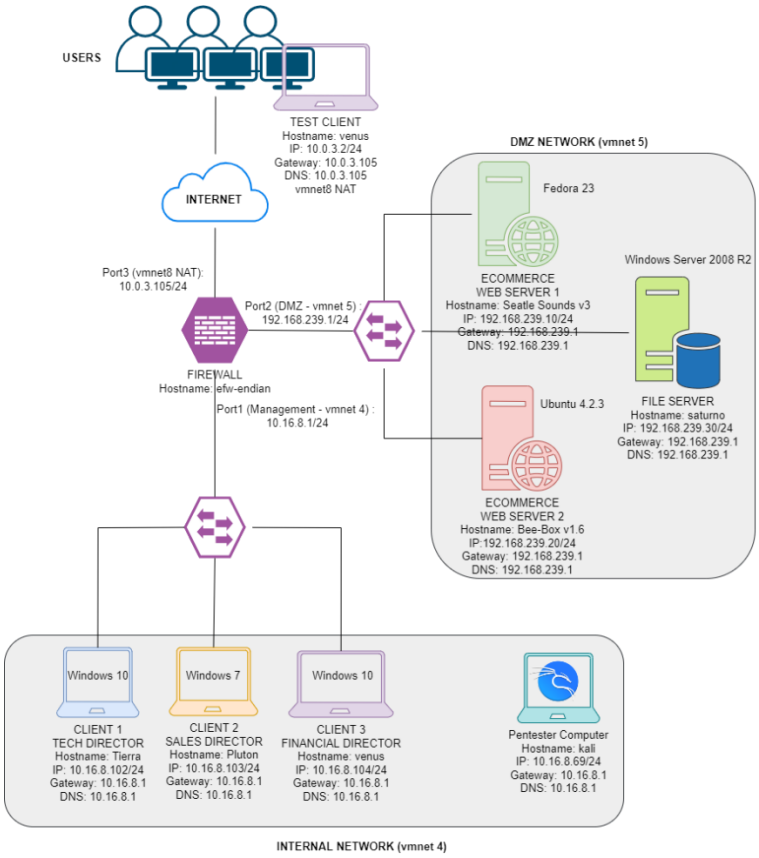
Lo que se hace es ignorar los puntajes de 0.0, que normalmente son solo de información, se suman todos los puntajes restantes y se obtiene el promedio de los hallazgos encontrados.

Es importante aclarar que, para este piloto, optamos por métodos directos y sencillos para calcular el riesgo combinando todas las fuentes de información. Sin embargo, en trabajos futuros puede definirse un método más conveniente para determinar los valores finales que se entregan al algoritmo FAIR. Además, vale la pena explorar otros algoritmos que puedan ofrecer una mayor fiabilidad en los cálculos realizados, con el objetivo de minimizar cualquier sesgo.

Finalmente, el procedimiento concluye con la aplicación de la fórmula específica para cada factor de riesgo (nodo) [55,56], en la que se calcula ya sea el promedio o la suma total de todas las entradas. Este paso se lleva a cabo con la finalidad de proporcionar el dato final a PyFAIR, lo cual posibilita su ejecución y la posterior simulación utilizando el método de Monte Carlo.

X. RESULTADOS

Con el objetivo de evaluar la herramienta ARASME fue implementado un ambiente de pruebas a través de un entorno virtualizado, que busca emular la infraestructura de una empresa PYME, y cuya arquitectura de red se muestra a continuación:



Los resultados obtenidos con la herramienta ARASME fueron contrastados con los obtenidos en la ejecución de un *risk assessment* manual, realizado por la empresa de seguridad PCA Ingeniería (https://pcaingenieria.com/site/), quien evaluó con su experto de ciberseguridad, utilizando la metodología MAGERIT, el mismo ambiente virtual de pruebas. Los resultados del *Risk assessment* manual realizado por la empresa PCA ingeniería y los obtenidos mediante la herramienta automática ARASME, para los dos (2) escenarios de riesgo propuestos, son los siguientes:

Prioridad	Escenario de Riesgo	Clasificación Riesgo Inherente				Clasificación Riesgo Residual			
		Puntaje MAGERIT (De 1 a 25)	Pérdida FAIR ($)			Puntaje MAGERIT (De 1 a 25)	Pérdida FAIR ($)		
			Mínima	Más probable	Máxima		Mínima	Más probable	Máxima
1°	Acceso no autorizado a la información	20	37 M	46 M	57 M	8	17 M	21 M	26 M
2°	Inyección SQL	16	28 M	35 M	43 M	4	7 M	13 M	16 M

Se observa que usando la metodología MAGERIT, en el *risk assessment* manual, cada riesgo es evaluado mediante un puntaje que va desde el valor de 1 (bajo riesgo) hasta 25 (máximo riesgo), mientras que usando la metodología FAIR en el *risk assessment* automático, el valor de riesgo calculado por la herramienta ARASME se expresa en términos monetarios ($COP) y se encuentra dentro de un rango de valores definido por una distribución de probabilidad tipo PERT [33], con un valor mínimo, máximo y más probable de pérdidas económicas. Es así, que la metodología FAIR, integrando múltiples fuentes de información, que incluyen fuentes OSINT y herramientas de escaneo, y los datos de entrada de usuario (suministrados a través de los formularios) permite asociar al riesgo un valor con contexto de negocio asociado a la organización evaluada.

Comparando los resultados se observa que el nivel de priorización de riesgo obtenido con los métodos manual y automático es exactamente el mismo, mostrando que el riesgo de mayor prioridad y, por ende, con mayor puntuación (y pérdidas monetarias), es el asociado al escenario de "Acceso no autorizado a la información", y el de menor prioridad es el riesgo de "inyección SQL".

Por otro lado, en el método manual y automático, se observa que la puntuación (y pérdidas monetarias) del riesgo residual es inferior al del riesgo inherente, para los dos (2) escenarios de análisis. Este resultado tiene sentido, debido a que el riesgo residual tiene en cuenta la aplicación de controles de seguridad en la organización, reduciendo el nivel de riesgo en los escenarios analizados.

XI. CONCLUSIONES Y TRABAJO FUTURO

El futuro de ARASME como aplicación ofrece un horizonte amplio y prometedor. Gracias a su diseño modular, ARASME puede expandirse e implementar diversas herramientas de seguridad, proporcionando un potencial ilimitado para abordar cualquier tipo de vulnerabilidad.

Además, ARASME ofrece la posibilidad de integrar metodologías adicionales de análisis de riesgos. Estas pueden combinarse o utilizarse independientemente para obtener resultados más precisos y efectivos.

Una posibilidad es su migración a la nube, lo que permitiría el uso de métodos y herramientas de seguridad basados en esta tecnología. Esto abre nuevas oportunidades para fortalecer aún más la seguridad de los sistemas.

Con la incorporación de inteligencia artificial, ARASME podría analizar de manera automática la información extraída de los análisis, generando así un proceso de evaluación más sofisticado y preciso. Esto, combinado con la capacidad de aumentar el flujo de información hacia recomendaciones de mejora y hasta la implementación correctiva y resolución de vulnerabilidades, hace de ARASME una herramienta de gran potencial.

En resumen, el potencial de la herramienta software ARASME es enorme, y representa una gran contribución al dominio de la identificación, priorización y evaluación de riesgos de la seguridad de la información, para empresas PYMES.

REFERENCIAS

[1] Small Business Administration (SBA). https://advocacy.sba.gov/2022/04/26/small-business-facts-small-business-job-creation/

[2] The majority of SMEs (>80%) process critical information, making cybersecurity a key concern. CYBERSECURITY FOR SMES (2021). https://www.enisa.europa.eu/publications/enisa-report-cybersecurity-for-smes/@@download/fullReport

[3] Kabanda, Salah & Tanner, Maureen & Kent, Cameron. (2018). Exploring SME cybersecurity practices in developing countries. Journal of Organizational Computing and Electronic Commerce. 28. 269-282. 10.1080/10919392.2018.1484598

[4] 29% of midmarket companies say breaches cost them less than $100K. 20% say it costs $1,000,000-$2,499,999 https://www.cisco.com/c/dam/global/hr_hr/solutions/small-business/pdf/small-mighty-threat.pdf

[5] trends and forecasts 2020–2025. https://www.analysysmason.com/contentassets/da7b5be3b59e4aae83ef98e5387209fb/analysys_mason_smb_security_worldwide_sample_jan2021_ren04.pdf

[6] Victims use the attack as a wake-up call to better protect themselves. Digital Surveys SMBs. https://digital.com/51-of-small-business-admit-to-leaving-customer-data-unsecure/

[7] Ehrlich, M., Lukas, G., Trsek, H., Jasperneite, J., & Diedrich, C. (2022). Investigation of Resource Constraints for the Automation of Industrial Security *Risk assessment*s. 2022 IEEE 18th International Conference on Factory Communication Systems (WFCS), 1–8. https://doi.org/10.1109/WFCS53837.2022.9779174

[8] Han, C.-H., & Han, C. (2021). Semi-quantitative cybersecurity *risk assessment* by blockade and defense level analysis. Process Safety and Environmental Protection, 155, 306–316. https://doi.org/10.1016/j.psep.2021.09.028

[9] Shamala, P., Ahmad, R., Zolait, A., & Sedek, M. (2017). Integrating information quality dimensions into information security risk management (ISRM). Journal of Information Security and Applications, 36, 1–10. https://doi.org/10.1016/j.jisa.2017.07.004

[10] Sukumar, A., Mahdiraji, H. A., & Jafari-Sadeghi, V. (2023). Cyber *risk assessment* in small and medium-sized enterprises: A multilevel decision-making approach for small e-tailors. Risk Analysis. https://doi.org/10.1111/risa.14092

[11] Abazi, B., & Kő, A. (2019). Semi-automated Information Security *Risk assessment Framework* for Analyzing Enterprises Security Maturity Level. 375, 141–152. https://doi.org/10.1007/978-3-030-37632-1_13

[12] Sánchez-García, I. D., Mejía, J., & San Feliu Gilabert, T. (2022). Cybersecurity *Risk assessment*: A Systematic Mapping Review, Proposal, and Validation. Applied Sciences, 13(1), 395. https://doi.org/10.3390/app13010395

[13] Karoui, K. (2016). Security novel *risk assessment Framework* based on reversible metrics: a case study of DDoS attacks on an E-commerce web server: Security Novel *Risk assessment Framework* Based on Reversible

Metrics. International Journal of Network Management, 26(6), 553–578. https://doi.org/10.1002/nem.1956

[14] Eckhart, M., Ekelhart, A., & Weippl, E. (2022). Automated Security Risk Identification Using AutomationML-Based Engineering Data. IEEE Transactions on Dependable and Secure Computing, 19(3), 1655–1672. https://doi.org/10.1109/TDSC.2020.3033150

[15] Marysse, C. (2016) Structural adaptive façades, Ghent University Library. 2016. Page 17. Available at https://lib.ugent.be/en/catalog/rug01:002300621 (Accessed: February 15, 2023).

[16] "Introduction to Return on Security Investment Helping CERTs assessing the cost of (lack of) security" (2012) Introduction to Return on Security Investment. Available at: https://www.enisa.europa.eu/publications/introduction-to-return-on-security-investment/@@download/fullReport (Accessed: February 15, 2023).

[17] ANIF. (2020). Gran encuesta pyme nacional ii 2020. https://www.anif.com.co/encuesta-mipyme-de-anif/gran-encuesta-pyme-nacional/

[18] Ehrlich, M., Lukas, G., Trsek, H., Jasperneite, J., & Diedrich, C. (2022). Investigation of Resource Constraints for the Automation of Industrial Security *Risk assessment*s. 2022 IEEE 18th International Conference on Factory Communication Systems (WFCS), 1–8. https://doi.org/10.1109/WFCS53837.2022.9779174

[19] Wang, Z., Chen, L., Song, S., Cong, P. X., & Ruan, Q. (2020). Automatic cyber security *risk assessment* based on fuzzy fractional ordinary differential equations. Alexandria Engineering Journal, 59(4), 2725–2731. https://doi.org/10.1016/j.aej.2020.05.014

[20] Eckhart, M., Ekelhart, A., & Weippl, E. (2022). Automated Security Risk Identification Using AutomationML-Based Engineering Data. IEEE Transactions on Dependable and Secure Computing, 19(3), 1655–1672. https://doi.org/10.1109/TDSC.2020.3033150

[21] Benz, M., & Chatterjee, D. (2020). Calculated risk? A cybersecurity evaluation tool for SMEs. Business Horizons, 63(4), 531–540. https://doi.org/10.1016/j.bushor.2020.03.010

[22] Armenia, S., Angelini, M., Nonino, F., Palombi, G., & Schlitzer, M. F. (2021). A dynamic simulation approach to support the evaluation of cyber risks and security investments in SMEs. Decision Support Systems, 147, 113580. https://doi.org/10.1016/j.dss.2021.113580

[23] Sukumar, A., Mahdiraji, H. A., & Jafari-Sadeghi, V. (2023). Cyber *risk assessment* in small and medium-sized enterprises: A multilevel decision-making approach for small e-tailors. Risk Analysis. https://doi.org/10.1111/risa.14092

[24] Tendencias de cibercrimen en Colombia. (2020). CCIT. Tendencias de Cibercrimen En Colombia, 36. https://www.ccit.org.co/wp-content/uploads/informe-tendencias-final.pdf

[25] Martins, B. F., Serrano Gil, L. J., Reyes Román, J. F., Panach, J. I., Pastor, O., Hadad, M., & Rochwerger, B. (2022). A *Framework* for conceptual characterization of ontologies and its application in the cybersecurity domain. Software and Systems Modeling, 21(4), 1437–1464. https://doi.org/10.1007/s10270-022-01013-0

[26] GEORGESCU, T. M., & SMEUREANU, I. (2017). Using Ontologies in Cybersecurity Field. Informatica Economica, 21(3/2017), 5–15. https://doi.org/10.12948/issn14531305/21.3.2017.01GEORGESCU, T. M., & SMEUREANU, I. (2017). Using Ontologies in Cybersecurity Field. Informatica Economica, 21(3/2017), 5–15. https://doi.org/10.12948/issn14531305/21.3.2017.01

[27] Grabis, J., & Bork, D. (2020). Conceptual Characterization of Cybersecurity Ontologies (Vol. 400). Springer International Publishing AG.

[28] Sikos, L. F. (2019). OWL Ontologies in Cybersecurity: Conceptual Modeling of Cyber-Knowledge (Vol. 151, pp. 1–17). Springer International Publishing AG. https://doi.org/10.1007/978-3-319-98842-9_1

[29] Poli, R., Healy, M., & Kameas, A. (2010). Theory and Applications of Ontology: Computer Applications (1st ed. 20). Springer Netherlands. https://doi.org/10.1007/978-90-481-8847-5

[30] Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., & Gangemi, A. (2012). Ontology Engineering in a Networked World (1st ed. 20). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-24794-1

[31] SERRANO, J. M. (2012). Applied Ontology Engineering in Cloud Services, Networks and Management Systems (1st ed. 20). Springer New York. https://doi.org/10.1007/978-1-4614-2236-5

[32] Möller, D. P. F. (2020). Cybersecurity Ontology (pp. 99–109). Springer International Publishing AG. https://doi.org/10.1007/978-3-030-60570-4_7

[33] Freund, J., & Jones, J. (2015). Measuring and managing information risk : a FAIR approach (1st editio). Butterworth-Heinemann.

[34] Wang, J., Neil, M., & Fenton, N. (2020). A Bayesian network approach for cybersecurity *risk assessment* implementing and extending the FAIR model. Computers & Security, 89, 101659. https://doi.org/10.1016/j.cose.2019.101659

[35] Chockalingam, S., Pieters, W., Herdeiro Teixeira, A. M., van Gelder, P. H. A. J. ., Lipmaa, H., Mitrokotsa, A., & Matulevicius, R. (2017). Bayesian Network Models in Cyber Security: A Systematic Review (Vol. 10674, pp. 105–122). Springer. https://doi.org/10.1007/978-3-319-70290-2_7

[36] Kelly, D. L., & Smith, C. L. (2009). Bayesian inference in probabilistic *risk assessment*—The current state of the art. Reliability Engineering & System Safety, 94(2), 628–643. https://doi.org/10.1016/j.ress.2008.07.002

[37] Fenton, N. E., & Neil, M. (Martin D. . (2013). *Risk assessment* and decision analysis with Bayesian networks (1st editio). CRC Press. https://doi.org/10.1201/b13102

[38] Mansfield-Devine, S. (2022). Sophos: The State of Ransomware 2022. Computer Fraud & Security, 2022(5). https://doi.org/10.12968/s1361-3723(22)70573-8

[39] ICCC FBI. (2021). Internet Crime Report 2021. Fedral Bureau of Investigation, Internet Crime Complaint Center, Washington, DC, 33. https://www.ic3.gov/Media/PDF/AnnualReport/2021_IC3Report.pdf

[40] Chidukwani, A., Zander, S., & Koutsakis, P. (2022). A Survey on the Cyber Security of Small-to-Medium Businesses: Challenges, Research Focus and Recommendations. IEEE Access, 10, 85701–85719. https://doi.org/10.1109/ACCESS.2022.3197899

[41] Crowdstrike, P. (2020). Realidades Ransomware para PYMES. https://www.ccit.org.co/wp-content/uploads/ransomware-para-pequenas-y-medianas-empresas-crowdstrike.pdf

[42] Gafni, R., & Pavel, T. (2019). The invisible hole of information on SMB's cybersecurity. Online Journal of Applied Knowledge Management, 7(1), 14–26. https://doi.org/10.36965/ojakm.2019.7(1)14-26

[43] Alberts, C., Dorofee, A., Stevens, J., & Woody, C. (2003). Introduction to the OCTAVE Approach. Pittsburgh, PA, Carnegie Mellon University, August.

[44] Alberts, C., Dorofee, A., Stevens, J., & Woody, C. (2005). OCTAVE-S Implementation Guide. Software Engineering Institute, 1(V 1.0), 1–63.

[45] Caralli, R. a R. a. C., Stevens, J. F., Young, L. R., & Wilson, W. R. (2007). Introducing OCTAVE Allegro : Improving the Information Security *Risk assessment* Process. Young, May, 1–113.

[46] Pinzón Guerrero, J. F., & Herrera Suescún, A. (2009). Acercamiento a la gestión de riesgos de TI con Magerit y las 4A . Uniandes.

[47] Syalim, A., Hori, Y., & Sakurai, K. (2009). Comparison of Risk Analysis Methods: Mehari, Magerit, NIST800-30 and Microsoft's Security Management Guide. 2009 International Conference on Availability, Reliability and Security, 726–731. https://doi.org/10.1109/ARES.2009.75

[48] Pastuszuk, J., Burek, P., & Księżopolski, B. (2021). Cybersecurity Ontology for Dynamic Analysis of IT Systems. Procedia Computer Science, 192, 1011–1020. https://doi.org/https://doi.org/10.1016/j.procs.2021.08.104

[49] Van Den Hooven -Issa Member, C., & Chapter, N. (2020). Quantitative Risk Calculation in Cybersecurity: The Value of Quantifying Risk Probability density function ISSA DEVELOPING AND CONNECTING CYBERSECURITY LEADERS GLOBALLY. October, 28–32. https://www.youtube.com/watch?v=EvHiee7gs9Y.

[50] Algarni, A. M., Thayananthan, V., & Malaiya, Y. K. (2021). Quantitative assessment of cybersecurity risks for mitigating data breaches in business systems. Applied Sciences, 11(8), 3678. https://doi.org/10.3390/app11083678

[51] Schafer, M., Fuchs, M., Strohmeier, M., Engel, M., Liechti, M., & Lenders, V. (2019). BlackWidow: Monitoring the Dark Web for Cyber Security Information. International Conference on Cyber Conflict, CYCON, 2019-May, 1–21. https://doi.org/10.23919/CYCON.2019.8756845

[52] Buron, M., Goasdoué, F., Manolescu, I., & Mugnier, M.-L. (2020). Obi-Wan: ontology-based RDF integration of heterogeneous data. Proceedings of the VLDB Endowment, 13(12), 2933–2936. https://doi.org/10.14778/3415478.3415512

[53] Calvanese, D., & De Giacomo, G. (2005). Data integration: a logic-based perspective. The AI Magazine, 26(1), 59–70

[54] 2023 FAIR Institute. https://www.fairinstitute.org/blog/3-key-concepts-in-fair

[55] The Mathematics of the Open FAIRTM Methodology. (n.d.). Retrieved June 1, 2023, from https://publications.opengroup.org/security-library/g224?_ga=2.152760825.1139537120.1685678503-686274908.1685678503

[56] Open FAIRTM Tool with SIPmathTM Distributions: Guide to the Theory of Operation. (n.d.). Retrieved June 1, 2023, from https://publications.opengroup.org/g181

[57] The Open FAIRTM Risk Analysis Tool (90-day Beta Evaluation License). (n.d.). Retrieved June 1, 2023, from https://publications.opengroup.org/i181?_ga=2.181509479.1139537120.1685678503-686274908.1685678503

[58] Open FAIRTM Risk Analysis Tool. (n.d.). Retrieved June 1, 2023, from https://publications.opengroup.org/q180?_ga=2.172031331.1139537120.1685678503-686274908.1685678503

[59] RiskLens | Cyber Risk Management. (n.d.). Retrieved June 1, 2023, from https://www.risklens.com/

[60] The Free Risk Analysis Training Application based on FAIR. (n.d.). Retrieved June 1, 2023, from https://www.fairinstitute.org/fair-u

[61] Welcome to pyfair's documentation! — pyfair 0.1-alpha.12 documentation. (n.d.). Retrieved June 1, 2023, from https://pyfair.readthedocs.io/en/latest/

[62] Walker, A., O'Connor, R. V, & Messnarz, R. (2019). RISKEE: A Risk-Tree Based Method for Assessing Risk in Cyber Security (Vol. 1060, pp. 45–56). Springer International Publishing AG. https://doi.org/10.1007/978-3-030-28005-5_4

[63] Enoch, S. Y., Ge, M., Hong, J. B., Alzaid, H., & Kim, D. S. (2018). A systematic evaluation of cybersecurity metrics for dynamic networks. Computer Networks (Amsterdam, Netherlands : 1999), 144, 216–229. https://doi.org/10.1016/j.comnet.2018.07.028

[64] Syed, Z., Pädia, A., Finin, T., Mathews, L., & Joshi, A. (2016). UCO: A Unified Cybersecurity Ontology. AAAI Workshop - Technical Report, WS-16-01-WS-16-15(Figure 1), 195–202.

[65] Syed, R. (2020). Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system. Information & Management, 57(6), 103334. https://doi.org/10.1016/j.im.2020.103334

[66] Sánchez-Zas, C., Villagrá, V. A., Vega-Barbas, M., Larriva-Novo, X., Moreno, J. I., & Berrocal, J. (2023). Ontology-based approach to real-time risk management and cyber-situational awareness. Future Generation Computer Systems, 141, 462–472. https://doi.org/10.1016/j.future.2022.12.006

[67] Alanen, J., Linnosmaa, J., Malm, T., Papakonstantinou, N., Ahonen, T., Heikkilä, E., & Tiusanen, R. (2022). Hybrid ontology for safety, security, and dependability *risk assessment*s and Security Threat Analysis (STA) method for industrial control systems. Reliability Engineering & System Safety, 220, 108270. https://doi.org/10.1016/j.ress.2021.108270

[68] Arogundade, O. T., Abayomi-Alli, A., & Misra, S. (2020). An Ontology-Based Security Risk Management Model for Information Systems. Arabian Journal for Science and Engineering, 45(8), 6183–6198. https://doi.org/10.1007/s13369-020-04524-4

AUTORES

Primer Autor– John Garcia, M.Eng. de la Seguridad de la información, Universidad de los Andes jw.garcia@uniandes.edu.co.
Segundo Autor – Mauricio Morales, M.Eng. de la Seguridad de la información, Universidad de los Andes s.moraleso@uniandes.edu.co
Tercer autor – Sergio Mahecha, M.Eng. de la Seguridad de la información, Universidad de los Andes a.mahecham@uniandes.edu.co