JIT 2022 RECONQUISTA
JÓVENES INVESTIGADORES TECNOLÓGICOS
Facultad Regional Rosario
Universidad Tecnológica Nacional
UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL RECONQUISTA
UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL RESISTENCIA
UTN FACULTAD REGIONAL RAFAELA
UTN FRVT
UTN SANTA FE
3 Y 4 DE NOVIEMBRE DE 2022 - UTN FACULTAD REGIONAL RECONQUISTA

# Modelado de amenazas de aplicaciones basadas en la nube con herramientas de software Open Source y software propietario

# Threat Modeling for Cloud-based applications with Open-Source Software and Proprietary Software tools

**Martín Ariel Escowich**
Universidad Tecnológica Nacional - Facultad Regional Santa Fe
escowichmartin@gmail.com.

## Resumen

En el siguiente trabajo se aborda un estudio comparativo de diferentes herramientas de Modelado de Amenazas. Se aplica el modelado de amenazas a un caso estudio, en particular el modelado de un sistema de Campus Virtual universitario, a fin de realizar el análisis de las posibles vulnerabilidades de este sistema. Para ello, se seleccionaron herramientas open source propuestas por la fundación OWASP y una herramienta propietaria. El proceso de investigación consistió en estudiar las características de ambas herramientas, definir los criterios de comparación, modelar el caso de estudio con las respectivas herramientas OWASP Threat Dragon/OdTM (como alternativa Open Source), y Threat Modelling Tool de Microsoft (como alternativa propietaria) y aplicar a esos modelos un razonador para detectar las posibles amenazas del sistema modelado.

## Palabras Clave

Modelado de amenazas, STRIDE, Ciberseguridad, Proceso de Desarrollo de Software

## Abstract

In this work, a comparative study of different Threat Modeling tools is addressed. Threat modeling is applied to a case study, in particular the modeling of a university Virtual Campus system, in order to carry out the analysis of the possible vulnerabilities of this system. For this, open source tools proposed by the OWASP foundation and a proprietary tool were selected. The research process consisted of studying the features provided by both tools, defining a comparison criteria, modeling the case study with OWASP Threat Dragon/OdTM (as an Open Source alternative), and Microsoft's Threat Modeling Tool (as a proprietary alternative), and apply a reasoner to these models to detect the possible threats of the modeled system.

## Keywords

Threat modeling, STRIDE, Cybersecurity, Software Development Process.

## Introducción

Los procesos de desarrollo de software tradicionales (como son el modelo en cascada y el modelo en espiral) contemplan cuatro etapas, una etapa inicial de análisis del contexto del sistema y los requerimientos de éste, una etapa de desarrollo, una etapa de prueba del comportamiento del sistema para que éste sea correcto, y una última etapa de despliegue y mantenimiento del sistema. Cada una de estas etapas se implementan de forma diferente en los distintos modelos de proceso de desarrollo, pero tienen

en común que apuntan a desarrollar un sistema que sea correcto y robusto ante errores. Sin embargo, en estos modelos, la seguridad del sistema es relegada a un segundo plano o no es considerada.

En la actualidad, es muy probable que los sistemas que se desarrollen funcionen en dispositivos conectados a la red, lo que abre la puerta a la interacción con usuarios maliciosos, por lo que es de vital importancia seguir un proceso de desarrollo seguro (esto es, un proceso que contemple las necesidades de seguridad del sistema desde el análisis hasta el despliegue y mantenimiento del mismo). Una de las prácticas recomendadas para desarrollo seguro es el modelado de amenazas (Shostack, 2014 [1]), tema que es el foco de este trabajo.

El modelado de amenazas es un proceso que se puede aplicar a cualquier entidad o proceso del sistema, pudiéndose tomar un alcance tan amplio como se desee. El modelado de amenazas consiste en definir los requisitos de seguridad, crear un diagrama de las entidades que conforman un sistema o software, identificar las amenazas, encontrar formas de mitigarlas, y validar las propuestas de mitigación (Figura 1).

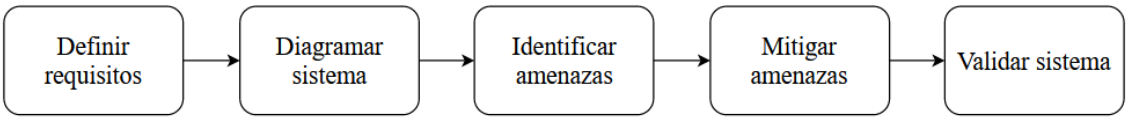



**Figura 1 - Proceso de Modelado de Amenazas**

Uno de los modelos de amenazas más populares es STRIDE (Shostack, 2014 [1]) para la identificación de amenazas de seguridad informática. Este modelo proporciona una sigla mnemotécnica para las amenazas a la seguridad en seis categorías: (S) Suplantación de identidad de usuario o Spoofing, (T) Tampering o manipulación de datos, (R) Repudio, (I) Information disclosure o divulgación de información (filtración de información), (D) Denegación de servicio, y (E) Elevación de privilegios. STRIDE es un modelo de amenazas, que, implementado en una herramienta, es útil para ayudar a razonar y encontrar amenazas a un sistema. Con una herramienta es posible describir un sistema, en función de procesos, bancos de datos, flujos de datos y fronteras de confianza. El modelado de amenazas es una metodología que permite responder a la pregunta "¿Qué puede salir mal en el sistema en el que estamos trabajando?", antes de avanzar en su desarrollo e implementación.

Cada amenaza es una violación de una propiedad deseable para un sistema: el Spoofing amenaza a la Autenticidad, Tampering a la Integridad, Repudio a la capacidad de No repudio, la Revelación de información a Confidencialidad, la Denegación de Servicio atenta a la Disponibilidad, y la Elevación de Privilegio es una amenaza a la capacidad de Autorización.

El objetivo de este trabajo es realizar un estudio comparativo de diferentes herramientas para modelar un sistema e identificar las posibles amenazas a las que el sistema puede ser vulnerable. Para ello, se seleccionan herramientas de software open source y propietarias, se estudiarán las características y funcionalidades de las diferentes herramientas, se definirá un conjunto de criterios de comparación, y se procederá a modelar un caso de estudio con las respectivas herramientas. Luego, se realizará una

evaluación del potencial que tiene cada herramienta para descubrir y alertar sobre posibles vulnerabilidades y si proveen de funcionalidades para sugerir mitigaciones de las mismas.

## Caso de estudio

El caso de estudio sobre el que se trabajó con las herramientas es un sistema de Campus Virtual universitario, compuesto por dos aplicaciones en la nube conectadas a un clúster de bases de datos. En la Figura 2, se presenta la arquitectura de componentes del mismo.

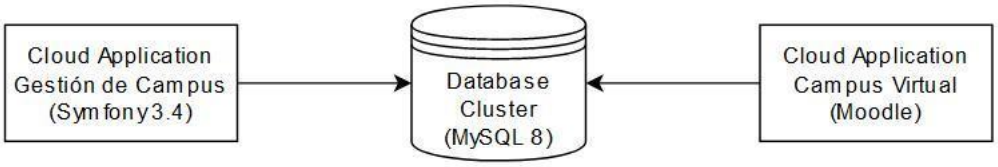



**Figura 2 - Arquitectura del sistema**

El sistema está compuesto por una aplicación en la nube destinada a la gestión del campus (creación de cursos y recursos), la cual está basada en el framework Symfony (Symfony, 2022 [2]) para generación de aplicaciones en lenguaje PHP, y por la plataforma del Campus Virtual, basada en Moodle (Moodle, 2022 [3]). Este caso de estudio es mayormente ilustrativo, y no se entrará en detalle a analizar las vulnerabilidades de las tecnologías específicas utilizadas para el sistema (es decir, no se analizarán vulnerabilidades conocidas de Moodle o Symfony). Ambos subsistemas trabajan sobre una base de datos MySQL 8, que se encuentra desplegada en la misma red.

## Metodología

Las herramientas seleccionadas para realizar la comparación fueron dos. Por un lado, considerando software open source, se trabajó con un set de herramientas que son parte del proyecto OWASP (OWASP, 2022 [4]). Ellas son Threat Dragon, como herramienta de modelado, y OWASP OdTM Framework para identificar amenazas. Por otro lado, la alternativa propietaria es la herramienta Microsoft Threat Modelling Tool, la cual permite realizar tanto modelado como identificación de amenazas.

Se emplearon las herramientas seleccionadas para modelar el caso de estudio y detectar las posibles vulnerabilidades que existen en el sistema. En particular, el interés de la evaluación es la usabilidad de las herramientas y no en la correctitud de los resultados arrojados en cuanto a vulnerabilidades se refiere, ya que ambas emplean catálogos de amenazas diferentes, y estos catálogos escapan al alcance del trabajo.

Los criterios considerados para la evaluación son:

- Facilidad de uso y UX (User eXperience)
- Versatilidad y Portabilidad
- Documentación

- Herramientas Open Source

Para trabajar con estas herramientas en primer lugar se requiere emplear la herramienta Threat Dragon, para generar un diagrama de flujo de datos (DFD) a partir de la arquitectura del sistema que se analizó (Figura 2). Este diagrama es empleado como entrada de la herramienta OdTM. Esta herramienta, hace uso de procedimientos de razonamiento automático para construir un modelo de amenazas del sistema (es decir, descubrir amenazas y contramedidas relevantes).

El enfoque de OdTM se basa en un modelo básico de amenazas que se encuentra implementado en una ontología. El modelo base permite la creación de varios modelos de amenazas específicos de dominio y su integración con fuentes externas, como enumeraciones de ataque/vulnerabilidad/debilidad, también con catálogos de patrones de seguridad (o amenazas) tradicionales. Para aplicar la herramienta en el caso de estudio se empleó el catálogo Academic Cloud Computing Threat Patterns (ATP), el cual es un modelo de amenazas específico de dominio. Es posible emplear un catálogo diferente específico a un dominio en particular. Todos los modelos se implementan como ontologías OWL (Web Ontology Language) con lógicas de descripción (DL) como base matemática (Barchuk, 2020 [5]; Barchuk, 2021 [6]).

El modelo básico permite la interpretación semántica de DFD (diagramas de flujo de datos) y la creación automática de listas de amenazas/contramedidas. Contiene conceptos e instancias para representar componentes de diagramas, amenazas, contramedidas y sus propiedades. Además, proporciona un enfoque STRIDE básico para el modelado de amenazas, el etiquetado de elementos de seguridad con diferentes etiquetas y perfiles de protocolo.

Para trabajar con OdTM, se necesita contar con el diagrama de flujo de datos (DFD) del sistema , un modelo de clases (las cuales serán asignadas a las entidades del DFD), un modelo base de amenazas, y un modelo de dominio específico (ambos en forma de ontologías .owl). Particularmente en esta evaluación, se utilizó como modelo de amenazas el catálogo ACCTP, que posee amenazas clasificadas según el perfil al que correspondan (Perfil de Arquitectura, de cumplimiento o de IaaS).

El modelo generado con Threat Dragon se construye con objetos básicos (actores, procesos y depósitos de datos) a los cuales se deben asociar clases específicas del modelo de amenazas considerado. El DFD en la Figura 3 es la implementación del caso de estudio en Threat Dragon. Threat Dragon permite mediante una interfaz ajustar la clase de cada entidad, y definir algunos parámetros dependientes de si la entidad es un actor, proceso o depósito de datos (Figura 4). Se puede observar que las entidades que se definieron se agrupan en las siguientes clases: *class#RemoteUser* para las entidades “Usuario Remoto” y “Administrador Campus”; *class#CloudApplication* clase de las entidades “Campus Virtual Moodle”, “DB Cluster” y “Sistema de Gestión de Campus”.

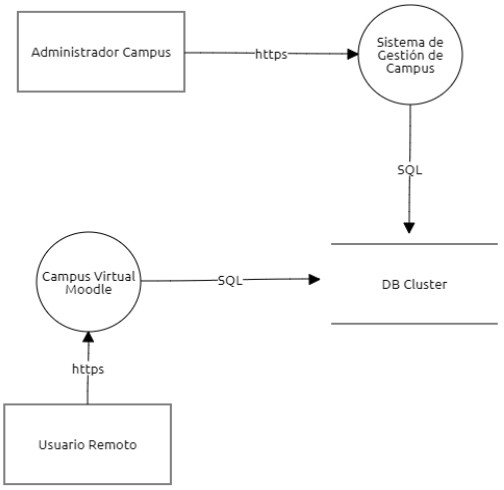


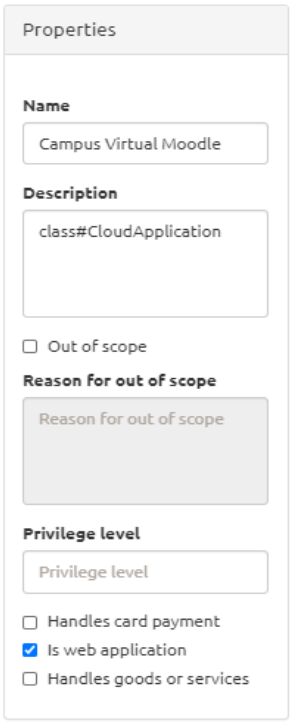

**Figura 3 - DFD del sistema considerado en el caso de estudio**

El diagrama y las propiedades de las entidades se exportan en formato .json para ser utilizados como entrada por la herramienta OdTM server, la cual se ejecuta con un razonador y un conjunto de ontologías .owl que representan a los modelos de amenazas empleados, para inferir cuáles son las amenazas posibles. Para ello es necesario configurar un archivo de formato ".properties" donde se asignan los parámetros utilizados (modelo de amenazas, modelo dfd y modelo de amenazas de dominio específico).

El proceso de identificación de amenazas finaliza obteniendo como resultado un archivo .json compatible con el formato admitido por Threat Dragon. Al abrir este archivo en Threat Dragon, si se selecciona una entidad, se observa la lista de amenazas posibles detectadas (Figura 5), y es posible acceder a un menú de gestión de amenazas para gestionar el estado de la vulnerabilidad asociada (indicando si ya se resolvió o se tomaron acciones para mitigarla), y documentar lo que se considere necesario.

- Herramienta de software Propietario

Microsoft Threat Modelling Tool (MTM) presenta funcionalidades similares a la suite de herramientas Open Source, pero en lugar de encontrarse dividida en dos aplicaciones como en el caso anterior, todas las funcionalidades se encuentran incorporadas dentro de una misma aplicación. El proceso de modelado es similar al anterior, solo que aquí tenemos la posibilidad de modelar entidades específicas del ecosistema de tecnologías de Microsoft para aplicaciones en la nube (por ejemplo, además de poder modelar bases de datos genéricas, es posible modelar bases de datos implementadas con Azure). Se procedió a modelar el sistema del caso de estudio y se generó un informe de amenazas (Figura 6). Al igual que la herramienta anterior, también es posible cargar catálogos de clases y amenazas personalizados, en caso de que sea necesario utilizar un catálogo que se ajuste a un dominio de sistema específico. La generación del informe de amenazas es intuitiva, y puede exportarse en formato .csv o en formato .htm , por lo que no hay mayores dificultades a la hora de manipular estos informes de amenazas que en el caso de la herramienta anterior. Entre la información brindada MTM se indica el título de la amenaza, la categoría STRIDE, una descripción, y posibles mitigaciones recomendadas.



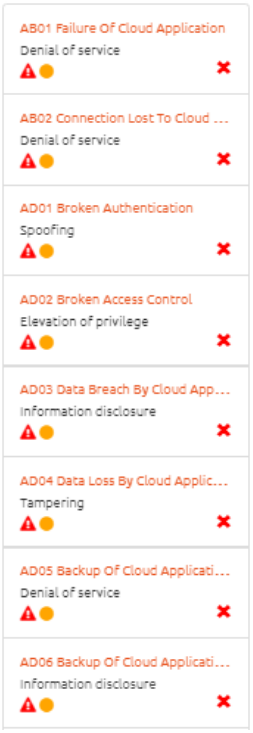
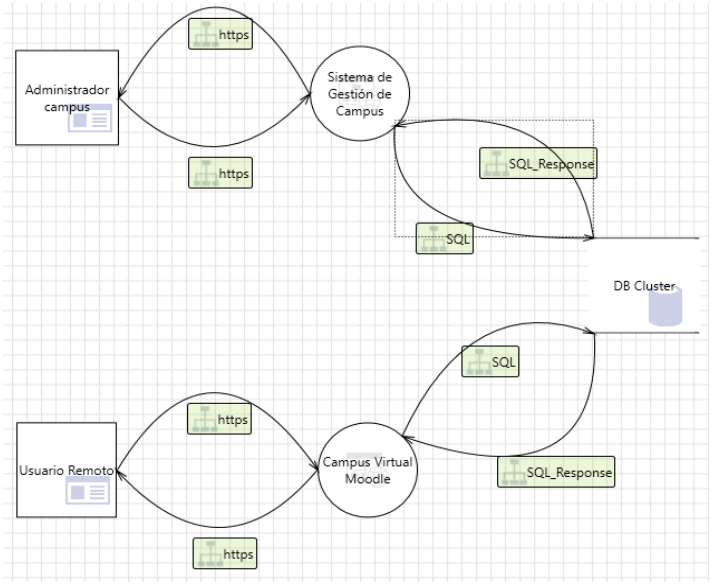




**Figura 4 - Interfaz de gestión de entidad** **Figura 5 - Amenazas detectadas para una entidad**



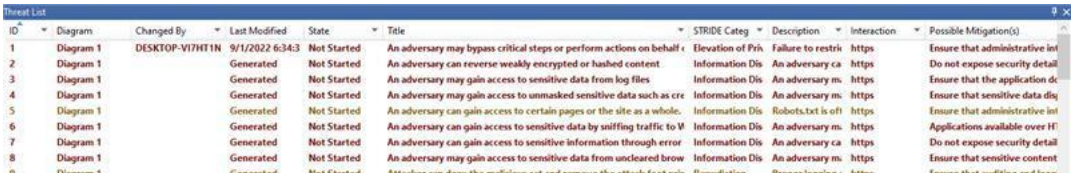




**Figura 6 - Diagrama modelado con Microsoft Threat Modelling Tool**

## Conclusiones

Como base de comparación de las herramientas empleadas, retomamos los criterios que fueron definidos. Con respecto a Facilidad de uso y UX, las herramientas Open Source son de instalación más difícil, en particular la aplicación de detección de amenazas OdTM server. El flujo de trabajo se ve interrumpido debido a que la configuración del programa para detectar amenazas en distintos modelos se realiza modificando manualmente un documento de texto. Por el contrario, la herramienta de Microsoft que permite modelar y luego detectar amenazas desde una sola ventana, con controles intuitivos.

En relación a Versatilidad y Portabilidad, las herramientas de OWASP pueden ejecutarse de forma nativa en Windows, MacOS y sistemas operativos basados en Linux, en tanto que la herramienta de Microsoft solo es compatible con Windows. En cuanto a los reportes que generan ambos programas, estos no están en un formato propietario, por lo que no presentan ningún inconveniente a la hora de convertir las salidas en documentación del Modelado de Amenazas.

En cuanto a la documentación disponible para aprender a usar las herramientas, se puede afirmar que existe amplia documentación para ambas aplicaciones, aunque la documentación de la aplicación Open Source supone que el usuario posee niveles de conocimiento avanzados. Esto implica que usuarios que no tengan experiencia ejecutando aplicaciones Java realizadas con Maven pueden tener dificultades para usar la herramienta. Por el contrario MTM provee documentación tanto sobre información la instalación y uso de la herramienta, como recomendaciones y guías para el desarrollo de software seguro. OWASP propone ejercicios de práctica para aprender a utilizar la herramienta que resultan útiles y logra cubrir la mayoría de las características de la aplicación.

En base al estudio realizado se concluye que al día de la fecha, para un desarrollador que quiera comenzar a trabajar con procesos de desarrollo seguros la herramienta de Microsoft es superior a la alternativa de OWASP, ya que la curva de aprendizaje es mucho menor, y provee por defecto la posibilidad de modelar y descubrir amenazas específicas para aplicaciones en la nube desarrolladas en el ecosistema Azure (un dato no menor, ya que Azure es uno de los mayores proveedores de servicios Cloud). El uso de las herramientas de OWASP es más afín al perfil de expertos en ciberseguridad que al perfil de un desarrollador, y el hecho de ser Open Source (que continúa en desarrollo) la hace más versátil y adaptable a dominios específicos y permite la posibilidad de extenderla.

## Referencias

[1] Shostack, Adam. Threat Modeling, Designing for Security. Wiley, 2014
[2] Symfony, https://github.com/symfony/symfony, accedido por última vez el 9/9/2022.
[3] Moodle, https://moodle.org/, accedido por última vez el 9/9/2022.
[4] Owasp, https://owasp.org/, accedido por última vez el 9/9/2022.
[5] Brazhuk A. “Security patterns based approach to automatically select mitigations in ontology-driven threat modeling” , Open Semantic Technologies for Intelligent Systems (OSTIS). – 2020. – №. 4. – С. 267-272
[6] Brazhuk A. “Threat modeling of cloud systems with ontological security pattern catalog”, International Journal of Open Information Technologies. – 2021. – Т. 9. – №. 5. – С. 36-41.