


UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

# Aplicación de Minería de Procesos para la Evaluación de Entrenamientos Ofensivos y Defensivos en Ciberseguridad

Guillermo Guerrero

Programa de Posgrado en Informática
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Agosto de 2025


UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

# Aplicación de Minería de Procesos para la Evaluación de Entrenamientos Ofensivos y Defensivos en Ciberseguridad

Guillermo Guerrero

Tesis de Maestría presentada al Programa de Posgrado en Informática, Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de Magíster en Informática.

Directores:
Dr. Gustavo Betarte
Dr. Juan Diego Campo

Director académico:
Dr. Juan Diego Campo

Montevideo – Uruguay
Agosto de 2025

Guerrero, Guillermo

Aplicación de Minería de Procesos para la Evaluación de Entrenamientos Ofensivos y Defensivos en Ciberseguridad / Guillermo Guerrero. - Montevideo: Universidad de la República, Facultad de Ingeniería, 2025.

XII, 100 p.: il.; 29, 7cm.

Directores:

Gustavo Betarte

Juan Diego Campo

Director académico:

Juan Diego Campo

Tesis de Maestría – Universidad de la República, Programa en Informática, 2025.

Referencias bibliográficas: p. 88 – 97.

1. Entrenamiento en Ciberseguridad, 2. Cyber Range, 3. Tectonic, 4. Evaluación, 5. Process Mining. I. Betarte, Gustavo, Campo, Juan Diego, . II. Universidad de la República, Programa de Posgrado en Informática. III. Título.

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS

---

Dr. Javier Baliosian

---

Dr. Lorena Etcheverry

---

Dr. Gerardo Simari

Montevideo – Uruguay
Agosto de 2025

# Agradecimientos

Deseo expresar mi más sincero agradecimiento a mis supervisores Dr. Gustavo Betarte y Dr. Juan Diego Campo por su constante apoyo y orientación a lo largo de la maestría. También agradezco a los miembros del tribunal por dedicar su tiempo y esfuerzo en la evaluación de este trabajo, en especial al Dr. Gerardo Simari, cuya revisión contribuyó significativamente a mejorar este documento. Finalmente, quiero agradecer a mi familia por su incondicional apoyo durante todo el trabajo de maestría.

# RESUMEN

En esta tesis se presenta una metodología para la evaluación del entrenamiento de usuarios en un *cyber range*. Los *cyber ranges* son plataformas que permiten simular entornos realistas para el entrenamiento práctico en el dominio de la ciberseguridad. La metodología propuesta utiliza como pilares una herramienta de SIEM, los *frameworks* MITRE ATT&CK y MITRE D3FEND, y la minería de procesos para el análisis, representación y estudio de las actividades llevadas a cabo por los participantes de los entrenamientos. Se realiza un estudio de otros trabajos relacionados así como un análisis comparativo de dichos trabajos con la metodología de evaluación propuesta, mostrando puntos de mejora e innovación introducidos por esta última. A su vez, se exponen casos de aplicación de la metodología para entrenamientos ofensivos y defensivos en el *cyber range* Tectonic, demostrando que, mediante el uso de esta metodología, el instructor logra identificar el cumplimiento de los objetivos de entrenamiento por parte de los participantes, y más importante aún, entender cómo estos realizan el entrenamiento.

Palabras claves:
Entrenamiento en Ciberseguridad, Cyber Range, Tectonic, Evaluación, Process Mining.

# Lista de figuras


2.1 Componentes de un *cyber range*. Extraído de [8]. . . . . 9
2.2 Tareas de la minería de procesos. Extraído de [33]. . . . . 14
2.3 *Frameworks* MITRE ATT&CK y D3FEND. Extraído de [43]. . 17

3.1 Funcionalidades claves de Tectonic . . . . . 22
3.2 Arquitectura de Tectonic . . . . . 28

4.1 Metodología de evaluación de entrenamientos en un *cyber range* 36
4.2 Ejemplo de modelo obtenido a raíz de la aplicación de descubrimiento de procesos. . . . . 41

5.1 Entrenamiento ofensivo - Modelo de referencia . . . . . 53
5.2 Escenario Ofensivo - Duración por grupo . . . . . 56
5.3 Entrenamiento ofensivo - Modelo del proceso de entrenamiento . 57
5.4 Entrenamiento ofensivo - Modelos de grupos 4 y 6 . . . . . 59
5.5 Entrenamiento ofensivo - Verificación de conformidad . . . . . 61
5.6 Escenario defensivo - Infraestructura del escenario . . . . . 64
5.7 Escenario defensivo - Modelo de referencia . . . . . 65
5.8 Escenario defensivo - Modelo del proceso de entrenamiento utilizando tácticas . . . . . 66
5.9 Escenario defensivo - Duración por grupo . . . . . 68
5.10 Escenario defensivo - Modelo del proceso de entrenamiento utilizando técnicas del MITRE D3FEND . . . . . 70
5.11 Escenario defensivo - Modelo del proceso de entrenamiento del grupo 6 . . . . . 71

A.1 Ejemplo modelo de proceso. Extraído de [102]. . . . . 100
A.2 Notación utilizada para los modelos. Extraído de [102]. . . . . 100

# Lista de tablas

3.1 Comparativa de *cyber ranges* . . . . . 30

4.1 Log de eventos de ejemplo . . . . . 39

5.1 Entrenamiento ofensivo - Distribución de eventos según la actividad . . . . . 54
5.2 Entrenamiento ofensivo - Cantidad de eventos por grupo . . . . 55
5.3 Escenario defensivo - Distribución de eventos según la actividad 66
5.4 Entrenamiento defensivo - Cantidad de eventos por grupo . . . . 67

# Lista de siglas

**AWS** Amazon Web Services 23, 52, 63

**BPMN** Business Process Modeling Notation 15

**CAR** Cyber Analytics Repository 44

**CERTuy** Centro Nacional de Respuesta a Incidentes de Seguridad Informática de Uruguay 1

**CLEI** Conferencia Latinoamericana de Informática 5, 35, 62

**CLI** Command Line Interface 13, 29, 31

**CTF** Capture The Flag 8, 33

**CTTP** Cyber Threat and Training Preparation 76

**DoS** Denial of Service 12

**ELI** Escuela Latinoamericana de Informática 62

**ENISA** European Union Agency for Cybersecurity 62

**FING** Facultad de Ingeniería 4

**FTP** File Transfer Protocol 52, 54, 63

**GCP** Google Cloud Platform 28

**GSI** Grupo de Seguridad Informática 4, 21, 30, 52, 62

**GUI** Graphical User Interface 13, 29, 31

**HTTP** Hypertext Transfer Protocol 27, 37, 51

**HTTPS** Hypertext Transfer Protocol Secure 24

**ICS** Industrial Control System 17

**IaC** Infrastructure as Code 28

**IaaS** Infrastructure as a Service 11

**LFI** Local File Inclusion 49

**LMS** Learning Management System 12, 29, 31

**LaSI** Laboratorio de Seguridad Informática 21

**MTD** Moving Tagret Defense 87

**PGP** Pretty Good Privacy 54

**PM** Process Mining 13, 14, 16, 37, 38, 40, 78

**RLMS** Range Learning Management System 9

**Red Ciberlac** Red de Excelencia en Ciberseguridad de Latinoamérica y el Caribe 21, 52

**SA** Situational Awareness 80

**SCADA** Supervisory Control and Data Acquisition 62

**SIEM** Security Information and Event Management 2, 4, 5, 13, 24, 26, 29, 33, 34, 36, 37, 38, 39, 43, 44, 45, 50, 51, 53, 54, 55, 60, 65, 74, 83, 85, 86

**SO** Sistema Operativo 37

**SQLi** Structured Query Language injection 32, 38

**SSH** Secure Shell 12, 29, 44, 52, 53, 54, 63

**TTP** Tacticas, Técnicas y Procedimientos 17

**UdelaR** Universidad de la República 4

# Tabla de contenidos

**Lista de figuras** VII

**Lista de tablas** VIII

**Lista de siglas** X

**1 Introducción** 1

**2 Fundamentos teóricos** 7

2.1 *Cyber ranges* . . . 7

2.1.1 Definición . . . 7

2.1.2 Componentes claves . . . 9

2.1.3 Exponentes de *cyber ranges* . . . 10

2.2 Process Mining . . . 13

2.3 *Frameworks* MITRE . . . 16

**3 Tectonic** 21

3.1 Contexto del proyecto . . . 21

3.2 Principales funcionalidades . . . 22

3.3 Arquitectura . . . 28

3.4 Comparativa con otras soluciones de *cyber range* . . . 30

3.5 Operativa de Tectonic . . . 32

**4 Metodología de Evaluación** 35

4.1 Propuesta de metodología . . . 35

4.1.1 Monitoreo y recolección de evidencias . . . 37

4.1.2 Descubrimiento . . . 40

4.1.3 Verificación . . . 42

4.2 Aplicación de la metodología según tipos de escenarios . . . 43

4.3 Casos de uso . . . . . 46
4.3.1 Análisis de entrenamiento de participantes . . . . . 46
4.3.2 Detección de trampa . . . . . 47
4.3.3 Análisis del diseño del escenario . . . . . 48

**5 Experimentación** **50**
5.1 Instanciación de la metodología en Tectonic . . . . . 50
5.2 Casos de aplicación . . . . . 52
5.2.1 Escenario ofensivo . . . . . 52
5.2.2 Escenario defensivo . . . . . 62

**6 Trabajos relacionados** **72**

**7 Conclusiones** **83**

**Referencias bibliográficas** **88**

**Anexos** **98**
Anexo 1 *Representación gráfica de modelos de procesos* . . . . . 99

# Capítulo 1

# Introducción

Con el avance de la tecnología y los medios digitales, la ciberseguridad es un dominio que ha tomado una gran relevancia en organizaciones e individuos con el fin de proteger a los datos de acceso no autorizado, alteraciones o pérdidas.

Los ataques cibernéticos perpetuados por actores maliciosos se encuentran al alza año tras año. Se estima que en 2025 estos ataques tendrán un costo total de 10,5 billones de dólares anuales a nivel global [1]. Esta es una realidad que afecta a todos los países, incluido Uruguay. Según estadísticas del Centro Nacional de Respuesta a Incidentes de Seguridad Informática de Uruguay (CERTuy), en 2024 se evidenciaron un total de 14.264 incidentes, cifra que viene en aumento de forma sostenida desde el año 2020. De los incidentes detectados en 2024, el 0,48 % son catalogados de severidad alta o muy alta. "*El costo promedio de su mitigación fue de U$S 75.000, lo que implica solo contener el impacto inmediato, sin incluir otros costos*" [2].

En definitiva, es necesario contar con profesionales del dominio que puedan hacer frente a esta realidad, asegurando los sistemas informáticos de organizaciones, instituciones y otras entidades. Sin embargo, una problemática evidenciada a nivel mundial es la falta de estos profesionales. Se estima que en 2025 existen 3,5 millones de puestos de trabajo en ciberseguridad sin cubrir a nivel de todo el mundo debido a la falta de recursos humanos en el área [3].

Por lo tanto, ante esta realidad es necesario formar y capacitar nuevos profesionales en ciberseguridad. Un aspecto muy importante de esta formación es que no puede ser pura y exclusivamente teórica. En ciberseguridad, al igual que en muchas otras áreas, el entrenamiento práctico es vital para adquirir y perfeccionar habilidades. Por tal razón es que surgieron los *cyber ranges*, como

plataformas que simulan infraestructuras para el entrenamiento práctico en ciberseguridad, permitiendo que los usuarios tomen tanto el rol de atacante como el rol de defensor.

Los entrenamientos que brindan estas plataformas se suelen realizar a través de escenarios. Los escenarios buscan transmitir ciertos conceptos de ciberseguridad, planteando un contexto de trabajo, objetivos y una infraestructura de entrenamiento. Es posible categorizar los escenarios en dos grandes tipos, según el rol que pueden cumplir los aprendices. Por un lado, están los entrenamientos ofensivos o *red team*, donde los usuarios cumplen el rol de un atacante y deben aplicar técnicas y emplear herramientas que les permitan atacar la infraestructura de entrenamiento. Por el contrario, están los entrenamientos defensivos o *blue team*, donde los usuarios toman el rol del defensor y su objetivo es defender la infraestructura de entrenamiento, identificando ataques y vulnerabilidades y aplicando controles de seguridad. Es importante que los aprendices se entrenen en ambos tipos de escenario para que adquieran conocimiento y habilidades en ambos aspectos de la materia, dado que para lograr una defensa adecuada y asegurar un sistema es importante tener conocimiento sobre los tipos de ataques a los cuales se enfrenta.

Una funcionalidad básica y a la vez muy importante en las plataformas de *cyber range* es la evaluación y seguimiento del entrenamiento de los participantes. De esta forma, el instructor puede evaluar el trabajo realizado por los participantes y comprender el grado de avance respecto a los objetivos de entrenamiento planteados en los escenarios. Nuestro trabajo realiza distintos aportes en la línea de la evaluación de entrenamiento en un *cyber range*.

En primer lugar, desarrollamos un estudio del estado del arte de otros trabajos que se ocupan de la evaluación de entrenamientos en estas plataformas. Esto nos permitió identificar: cómo realizan la evaluación otros autores, técnicas y herramientas aplicadas, puntos en común y posibles puntos de mejoras que pretendemos subsanar con nuestra propuesta.

Por otra parte, diseñamos una nueva metodología de evaluación de entrenamientos ofensivos y defensivos llevados a cabo en un *cyber range*. La metodología propuesta utiliza como pilares fundamentales una herramienta de Security Information and Event Management (SIEM), los *frameworks* MITRE ATT&CK y MITRE D3FEND y la minería de procesos para el estudio, análisis, representación y comunicación de las actividades llevadas a cabo por los participantes para el cumplimiento de los objetivos de entrenamiento. El SIEM

permite monitorear la infraestructura de entrenamiento para la obtención de eventos y registros de auditoría, y su posterior análisis y correlación en base a analíticas de seguridad con el objetivo de detectar e identificar las acciones clave, tanto de ataque como de defensa, que realizan los participantes como parte del entrenamiento. A su vez, permite etiquetar estas acciones en base a tácticas y técnicas de los *frameworks* de MITRE. Estos *frameworks* son útiles para la categorización de acciones de ataque y defensa en base a tácticas, objetivos de alto nivel, y técnicas, acciones específicas para cumplir con estos objetivos. Por último, la disciplina de minería de procesos posibilita el descubrimiento, análisis y estudio de procesos, entendiendo a un proceso como una secuencia de pasos ejecutados para cumplir con cierto fin.

Nuestra metodología permite la generación de modelos visuales que representan los procesos de ataque o defensa realizados por los participantes como parte de su entrenamiento. Estos procesos son generados a raíz de las actividades ejecutadas por los participantes que se encuentran descriptas en base a tácticas o técnicas de los *frameworks* MITRE ATT&CK y D3FEND. A su vez, es posible comparar la actividad ejecutada por los participantes con modelos de referencia que representan la actividad esperada para el cumplimiento de los objetivos de entrenamiento, con el fin de identificar desviaciones.

Mediante la aplicación de la metodología propuesta, los instructores pueden detectar el cumplimiento de los objetivos de entrenamiento por parte de los participantes, y más importante aún, comprender cómo los participantes realizan el entrenamiento, ya que es posible identificar actividades específicas llevadas a cabo y herramientas empleadas. El conocimiento adquirido por el instructor a raíz de la aplicación de esta metodología de evaluación puede ser utilizado para brindar *feedback* a los participantes, destacando tanto puntos positivos en su entrenamiento como puntos a mejorar y perfeccionar. A su vez, el instructor podría aprovechar esta metodología para otros dos fines: la detección de trampa por parte de los participantes y el estudio del diseño de los escenarios analizando cuestiones como el nivel de dificultad y formas de resolución imprevistas.

En definitiva, la metodología propuesta en nuestro trabajo plantea ciertas mejoras respecto a otros trabajos del área. En primer lugar, hacemos uso de los *frameworks* MITRE ATT&CK y MITRE D3FEND para la categorización de la actividad ejecutada por los participantes de los entrenamientos en tácticas y técnicas empleadas por atacantes y defensores. Esto da lugar a reutilizar

todo el conocimiento embebido en estos *frameworks* y expresar la actividad de los participantes en un lenguaje común que simplifica la comprensión por parte del instructor. Adicionalmente, es factible analizar la actividad de los participantes en distintos niveles de granularidad, siendo posible expresar su actividad y generar modelos a niveles de tácticas, que representan los objetivos a alto nivel, o a nivel de técnicas, las cuales permiten entender con mayor detalle cómo cumplen con esos objetivos. Por otra parte, mediante la aplicación de la metodología es posible descubrir modelos de referencia que detallen la actividad ideal de entrenamiento en base a la propia actividad llevada a cabo por los participantes, incluso en escenarios complejos donde esta tarea podría ser muy dificultosa para que un instructor la realice de forma manual. Estos modelos de referencia pueden ser enriquecidos con el conocimiento que aporta el instructor como experto en ciberseguridad y diseñador del escenario.

En términos generales, esta metodología no es solamente una construcción teórica, sino un enfoque práctico y sistemático que utiliza técnicas y tecnologías avanzadas de minería de procesos y análisis de seguridad, brindando al instructor la posibilidad de evaluar sesiones de entrenamiento en un *cyber range* e identificar posibles brechas y áreas de mejora, adquiriendo así una comprensión más profunda de los procesos involucrados en la realización de estas actividades de capacitación.

Para demostrar la viabilidad de la metodología, la aplicamos para la evaluación de sesiones de entrenamiento realizadas en el *cyber range* Tectonic [4]. Tectonic es un *cyber range* académico diseñado e implementado por el Grupo de Seguridad Informática (GSI) de la Facultad de Ingeniería (FING) de la Universidad de la República (UdelaR) utilizado para el dictado de múltiples cursos de grado y posgrado ofrecidos por el grupo en torno a la ciberseguridad. Para poder aplicar la metodología a estos casos de estudio, primero fue necesario extender las funcionalidades del *cyber range*. Como principales desarrolladores de la plataforma Tectonic, integramos dos tecnologías clave: Elastic Security, una tecnología de SIEM, y Caldera, un *framework* para la especificación e implementación de ataques automatizados sobre una infraestructura computacional. La primera herramienta, como fue mencionado previamente, permite el monitoreo de la actividad de los estudiantes en su entrenamiento, su análisis para la identificación de acciones ejecutadas, y la categorización de dicha actividad en base a los *frameworks* de MITRE. Además del uso más directo de la herramienta para el análisis de actividad ofensiva, en nuestro

trabajo proponemos utilizarla de una forma innovadora para el análisis de actividad defensiva. Es decir que este SIEM no solo permite detectar e identificar actividad maliciosa u ofensiva, sino que también actividad defensiva mediante el uso de analíticas de seguridad cuidadosamente diseñadas por el instructor que idea un escenario. En cambio, la segunda herramienta la utilizamos para la implementación de escenarios de tipo defensivo en el *cyber range*, dado que posibilita la ejecución de ataques automatizados a demanda ante los cuales los participantes de los entrenamientos deben responder.

En síntesis, nuestro trabajo realiza los siguientes aportes en la evaluación del entrenamiento de usuarios en un *cyber range*:

- Estudio de estado del arte sobre metodologías y técnicas empleadas por *cyber ranges* y otras plataformas para la evaluación de entrenamiento de usuarios en el dominio de la ciberseguridad.
- Propuesta de metodología para la evaluación del entrenamiento en plataformas de *cyber range* utilizando como pilares fundamentales una herramienta de SIEM, la minería de procesos y los *frameworks* MITRE ATT&CK y MITRE D3FEND.
- Extensión del *cyber range* Tectonic, incorporando las herramientas Elastic Security y Caldera.
- Aplicación de la metodología a distintos casos de entrenamiento llevados a cabo en Tectonic.
- Divulgación de las anteriores contribuciones mediante publicaciones en las conferencias: Conferencia Latinoamericana de Informática (CLEI) [5] y ARGENCON [6], sucedidas durante el año 2024. A raíz de la publicación en el CLEI fuimos invitados a generar una versión extendida del artículo para ser publicada en el CLEI Electronic Journal. Este nuevo artículo [7] fue aceptado pero aún no ha sido publicado.

Por último, el presente documento se encuentra estructurado de la siguiente forma. El Capítulo 2 presenta conceptos bases necesarios para la comprensión del trabajo. El *cyber range* Tectonic, utilizado como base para la aplicación de la metodología, se introduce en el Capítulo 3. La propuesta de metodología para la evaluación de entrenamientos en un *cyber range* se encuentra detallada en el Capítulo 4, mientras que el Capítulo 5 presenta casos de aplicación de esta metodología. Por último, en el Capítulo 6 se presentan trabajos relacionados del área y un análisis comparativo, mientras que en el Capítulo 7 se presentan

las conclusiones del trabajo y líneas de investigación futuras.

# Capítulo 2

# Fundamentos teóricos

En este capítulo se presentan conceptos previos que son necesarios para entender nuestro trabajo. En este sentido se introducen las plataformas de *cyber range*, así como el dominio de minería de procesos y los *frameworks* MITRE ATT&CK y MITRE D3FEND.

## 2.1. *Cyber ranges*

En esta sección se introducen las plataformas de *cyber ranges*, presentando su definición, componentes claves y ejemplos.

### 2.1.1. Definición

Los *cyber ranges* son plataformas que surgen con el objetivo de simular ambientes compuestos de redes, sistemas y aplicaciones, que permiten el entrenamiento de usuarios en temas afines a la ciberseguridad. La idea es que los usuarios pueden adquirir nuevas habilidades o afianzar conceptos previamente aprendidos, de una forma práctica, segura y legal [8]. En particular, las cualidades de seguridad y legalidad son de vital importancia ya que muchas veces los entrenamientos implican que los usuarios ataquen una infraestructura o practiquen con herramientas o *software* (por ejemplo *malware*) que pueden ocasionar un daño. En consecuencia, es relevante brindar ambientes aislados y seguros.

Estas plataformas pueden estar destinadas a distinto público objetivo. En primer lugar, se tiene a los educadores y docentes, quienes desean brindar cursos de entrenamiento en temas de ciberseguridad. Estos pueden implementar

entrenamientos prácticos que tratan problemáticas o conceptos previamente enseñados en el componente teórico de los cursos. Por otra parte, se tiene a los estudiantes y profesionales, quienes realizan estos entrenamientos para adquirir, desarrollar y mejorar sus habilidades. Por último, se tienen a las organizaciones que desean mejorar las habilidades y capacidad de respuesta de sus equipos de ciberseguridad.

Por lo tanto, estas plataformas tienen tres actores o roles distintos: los instructores, representados por los docentes o organizaciones que brindan los entrenamientos; los aprendices, que son los estudiantes o profesionales que realizan los entrenamientos; y por último, los administradores encargados de gestionar la plataforma y asegurar el correcto funcionamiento.

Se podría decir que las plataformas de *cyber range* constan de dos grandes pilares. Por una parte, se tiene la plataforma en sí, es decir, la herramienta que permite la implementación de los entrenamientos y la infraestructura sobre la cual se brindan estos entrenamientos. Por otra parte, están los escenarios de entrenamiento. Los escenarios buscan transmitir ciertos conceptos de ciberseguridad, planteando un contexto de trabajo y objetivos de entrenamiento. Además, describen la infraestructura y otros componentes que deben ser desplegados y configurados para que los usuarios puedan entrenarse.

En este sentido, es posible categorizar los escenarios en dos grandes tipos según el rol que cumplen los aprendices. Estas categorías suelen estar asociadas a un color de la denominada *infosec color wheel* [9, 10]. Por un lado, están los entrenamientos ofensivos o *red team*, donde los usuarios cumplen el rol de un atacante y deben aplicar técnicas y emplear herramientas que les permitan atacar la infraestructura de entrenamiento. Por el contrario, están los entrenamientos defensivos o *blue team*, donde los usuarios toman el rol del defensor y su objetivo es defender la infraestructura de entrenamiento, identificando ataques y vulnerabilidades y aplicando controles de seguridad. En el Capítulo 5 se presentarán ejemplos de ambos tipos de escenarios.

Es importante que los aprendices se entrenen en ambos tipos de escenario para que adquieran conocimiento y habilidades en ambos aspectos de la materia, dado que para lograr una defensa adecuada y asegurar un sistema es importante tener conocimiento sobre los tipos de ataques a los cuales se enfrenta.

Es muy común que los escenarios sigan una modalidad de tipo Capture The Flag (CTF). En esta modalidad, los usuarios deben realizar distintas tareas

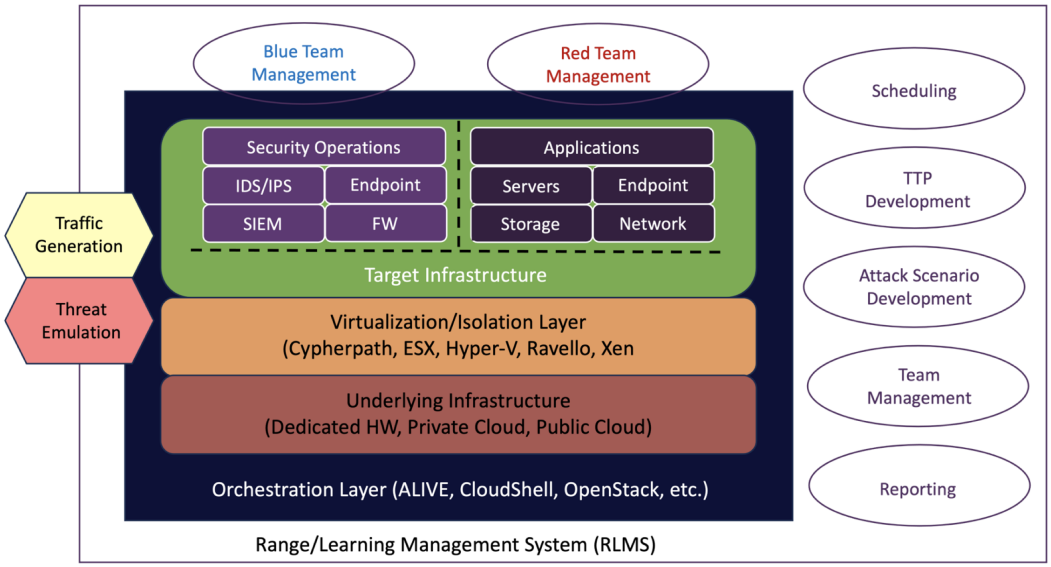



**Figura 2.1:** Componentes de un *cyber range*. Extraído de [8].

para cumplir con los objetivos de entrenamiento que generalmente implican obtener una o más *flags*. Las *flags* son esencialmente *strings* únicos que se suelen dejar en archivos que forman parte de las máquinas de entrenamiento y que los usuarios deben identificar y entregar como parte de la resolución del escenario. Si el aprendiz entrega las *flags* correctas entonces ha realizado las acciones necesarias para cumplir con el escenario de forma satisfactoria.

Adicionalmente existen escenarios más complejos donde, en la misma infraestructura de entrenamiento, conviven el *red* y *blue team*. El objetivo del *red team* es atacar la infraestructura mientras que el *blue team* realiza esfuerzos por mitigar y prevenir dichos ataques.

### 2.1.2. Componentes claves

Los componentes clave de alto nivel de un *cyber range* se encuentran representados en la Figura 2.1.

El componente Range Learning Management System (RLMS) permite la gestión, documentación, seguimiento y automatización de los escenarios de entrenamiento que brinda la plataforma. Algunas de las tareas encargadas de este componente son: definición de escenarios, recolección y reporte de datos sobre el entrenamiento de los aprendices y sobre el estado de funcionamiento de la propia plataforma.

La capa de orquestación es la encargada de gestionar el despliegue de la

infraestructura que conforma al escenario de entrenamiento, es decir, se encarga de la creación y destrucción de la infraestructura objetivo sobre la que se entrenan los aprendices.

Este despliegue se realiza sobre una infraestructura base, que puede ser tanto en servidores *on-premises* o en la *cloud*. Generalmente, los *cyber ranges* hacen uso de una capa de virtualización, la cual permite reducir costos asociados al despliegue sobre una infraestructura física, utilizando tecnologías como contenedores y máquinas virtuales. Hace varios años no era común encontrar infraestructuras computacionales implementadas con virtualización, por lo que el uso de esta capa en un *cyber range* podría afectar el realismo de los escenarios implementados. Sin embargo, en los últimos años, la virtualización ha sido adoptada ampliamente, tanto para la implementación de infraestructuras *on-premises* como en la nube. Las organizaciones y empresas suelen implementar su infraestructura utilizando una capa de virtualización y no suele ser común el despliegue de servidores físicos como tal. Por lo tanto, el uso de esta capa cobra sentido para la simulación de escenarios realistas de la actualidad.

Por último, los *cyber ranges* suelen tener capacidades o funciones extras que aporten al realismo de los entrenamientos. Generalmente se suelen tener funcionalidades que permiten la simulación de tráfico y la ejecución de ataques automatizados sobre la infraestructura de entrenamiento.

### 2.1.3. Exponentes de *cyber ranges*

En la actualidad existen múltiples soluciones de *cyber range*, cada una con sus características particulares o destinadas a un tipo de usuario en concreto. Algunos ejemplos de plataformas de *cyber ranges* que tienen foco en estudiantes son: Clusus [11], CyTrONE [12] y KYPO [13]. También existen *cyber ranges* destinados a profesionales y organizaciones como son: THREAT-ARREST [14], Cyberwiser.eu [15] y Cyberbit [16]. Incluso existen *cyber ranges* destinados a todo público como pueden ser TryHackMe [17] y Hack The Box [18]. Acto seguido se exponen brevemente algunas de estas plataformas, destacando funcionalidades y componentes clave. En la sección 3.4 se realiza un análisis comparativo de estas plataformas con la plataforma Tectonic.

En primer lugar, es necesario contar con un lenguaje que permita especificar escenarios de entrenamiento y los distintos componentes que los conforman. Generalmente, cada *cyber range* suele definir un lenguaje propio. El *cyber range*

CyTrONE define un lenguaje basado en YAML [19] donde es posible especificar las máquinas y redes que conforman el escenario, así como configuraciones que deben ser aplicadas sobre las máquinas. El lenguaje le ofrece al instructor un conjunto de acciones de configuración que puede utilizar. Estas acciones son: agregar usuarios, instalar paquetes, configurar reglas de *firewall*, copiar contenido a una máquina y ejecutar *scripts*. Por otra parte, KYPO también cuenta con su propia definición de escenarios. De forma similar al caso anterior, el lenguaje se basa en YAML y permite definir máquinas y topología de red a desplegar. Adicionalmente, permite definir *routers* que forman parte de la infraestructura del escenario. Sin embargo, la gran diferencia respecto a CyTrONE es que las configuraciones de máquinas y *routers* se realizan a través de *playbooks* Ansible [20] que el instructor debe proveer. Ansible es una de las herramientas de gestión de configuraciones más utilizadas en la actualidad. Si bien es necesario conocer Ansible para definir estos *playbooks*, es relativamente sencillo aprender a utilizar la herramienta. El uso de Ansible facilita considerablemente las tareas de configuración que deben ser aplicadas sobre una máquina, ya que permite su automatización y asegura (en cierta medida) que los resultados sean consistentes. Esto se debe a la condición de idempotencia de las tareas de Ansible. Una tarea es idempotente si el resultado de aplicarla una vez es el mismo que al aplicarla repetidas veces [21]. Es decir, que ante la aplicación de una misma configuración siempre se obtiene el mismo resultado. El *cyber range* THREAT-ARREST hace uso de una especificación de escenarios (no fue posible hallar y acceder a documentación que describa detalles particulares) que se encuentra empaquetada como parte de un programa de entrenamiento. Este programa define, la infraestructura de entrenamiento, objetivos de entrenamiento, acciones que se espera que realice el usuario para cumplir con el entrenamiento, por ejemplo, ataques que deben ser realizados sobre la infraestructura, y otros detalles sobre qué componentes de la infraestructura deben ser monitorizados para la posterior evaluación del entrenamiento.

Por otra parte, en lo que refiere a la gestión del escenario, y en particular su despliegue, cada *cyber range* suele contar con distintas tecnologías de virtualización o de Infrastructure as a Service (IaaS) que permiten el despliegue de forma local, en la nube o en ambientes híbridos. Por ejemplo, Clusus hace uso de tecnologías como Docker [22] y Virtualbox [23] para la generación de contenedores y máquinas virtuales que pueden ser desplegadas tanto en nubes

públicas como de forma local. CyTrONE, a través de su módulo CyRIS [24] permite el despliegue de máquinas virtuales en servidores locales utilizando la tecnología de virtualización Libvirt [25] o en la nube pública AWS [26]. En cambio, KYPO utiliza Packer [27] y OpenStack [28] para la generación de máquinas y su despliegue. Por último, CyberWiser.eu, además de permitir el despliegue en AWS o OpenStack, también incorpora el despliegue con la tecnología OpenNebula [29]. Tanto OpenStack como OpenNebula son tecnologías que permiten implementar infraestructuras *cloud* siguiendo el paradigma IaaS.

Otro de los componentes clave en los *cyber ranges* es aquel que permite la simulación de ataques sobre el escenario. CyTrONE ofrece esta funcionalidad a través de su módulo CyPROM [30], permitiendo la ejecución de ataques de fuerza bruta[1] al servicio Secure Shell (SSH) o ataques de tipo Denial of Service (DoS)[2]. Adicionalmente, permite ejecutar programas básicos sobre las máquinas que conforman un escenario con el objetivo de simular *malware*. Otros *cyber ranges* como THREAT-ARREST y CyberWiser.eu, indican que permiten la ejecución de ataques automáticos, pero no fue posible obtener detalles sobre su implementación o tipos de ataques permitidos ya que la documentación a la que se tuvo acceso no lo aclara.

Es deseable que los *cyber ranges* ofrezcan funcionalidades que permitan la evaluación del entrenamiento de los participantes. Esta evaluación puede ser realizada de varias formas. Por un lado, en base a cuestionarios que los participantes deben completar, pudiendo ser previos al entrenamiento para identificar su nivel inicial o posteriores al entrenamiento para evaluar que efectivamente han adquirido y entendido los conceptos tratados. Los *cyber ranges* KYPO, Clusus y CyTrONE (particularmente con su módulo CyLMS [31]) siguen este enfoque. El caso de estos dos últimos *cyber ranges* es similar ya que se integran con el Learning Management System (LMS) Moodle [32] para realizar estos cuestionarios. En cambio, otra forma de evaluación puede ser a través de las *flags*. Esto le permite al instructor identificar que los participantes efectivamente han completen el escenario de entrenamiento. Los *cyber ranges* KYPO, Clusus y CyTrONE también incorporan esta metodología de evaluación. Adicionalmente, otra metodología de evaluación puede ser en base a eventos significativos realizados por los participantes durante su entrena-

[1]Los ataques de fuerza bruta implican que un actor malicioso ejecuta un proceso de prueba y error para la obtención de credenciales para la autenticación en un sistema.

[2]Los ataques de denegación de servicio tienen como objetivo afectar la normal operativa de un servicio haciendo que este sea inaccesible para los usuarios legítimos.

miento. KYPO, Clusus, THREAT-ARREST y CyberWiser.eu hacen uso de esta metodología ya que permiten recolectar eventos desde la infraestructura de entrenamiento, procesarlos y presentarlos al instructor para que este pueda comprender el avance de los participantes. En el caso particular de CyberWiser.eu fue posible obtener más detalle sobre su implementación y se conoce que utiliza una tecnología de SIEM para el análisis y correlación en tiempo real de eventos obtenidos desde la infraestructura de entrenamiento para la detección e identificación de ataques ejecutados por los participantes. Tanto THREAT-ARREST como CyberWiser.eu, permiten analizar la infraestructura de entrenamiento en busca de vulnerabilidades y vectores de ataque. Esta forma de evaluación permitiría, una vez cumplido el entrenamiento, determinar la correcta resolución de vulnerabilidades por parte de los participantes que cumplen el rol de defensor en entrenamientos de tipo *blue team*.

Por último, los *cyber ranges* suelen proveer distintas interfaces para los distintos usuarios de la plataforma. Centrándose en las interfaces ofrecidas a los instructores, las distintas plataformas como KYPO, Clusus, THREAT-ARREST y CyberWiser.eu ofrecen interfaces de tipo Graphical User Interface (GUI), en cambio CyTrONE ofrece interfaces de tipo Command Line Interface (CLI).

## 2.2. Process Mining

Process Mining (PM) es una disciplina que tiene como objeto de estudio y análisis a los procesos. Se entiende por proceso a una secuencia de actividades que se ejecutan en cierto orden para cumplir con un objetivo determinado. El proceso bajo estudio puede ser de cualquier índole. A modo de ejemplo, se puede trabajar con el proceso asociado a cómo se atienden pacientes en un hospital o el proceso asociado a la línea de ensamblaje de un automóvil en una fábrica. Lo interesante de PM es que permite estudiar procesos reales, es decir procesos que efectivamente ocurren y se ejecutan en cierto contexto.

En el contexto de un *cyber range*, el entrenamiento realizado por los usuarios de estas plataformas puede ser interpretado como un proceso, dado que los participantes desempeñan distintas acciones de defensa o ataque sobre una infraestructura de entrenamiento. Por ejemplo, se pueden utilizar herramientas de ataque para obtener credenciales de acceso las cuales permitan un acceso indebido a una aplicación o sistema. Estas tareas suelen ser aplicadas siguien-

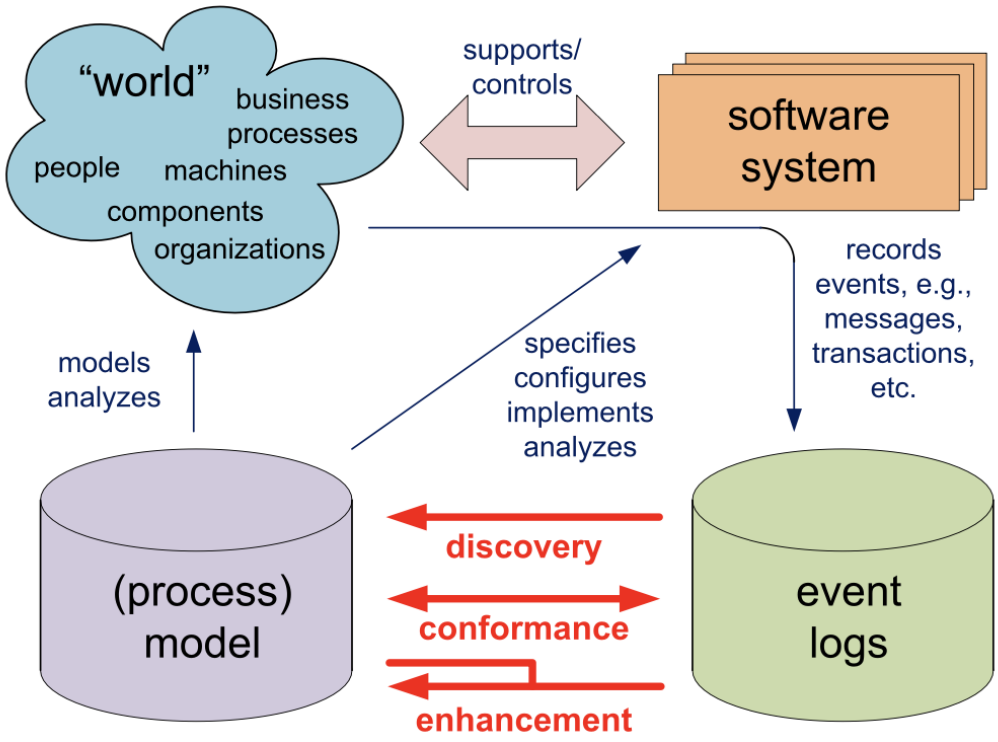



**Figura 2.2:** Tareas de la minería de procesos. Extraído de [33].

do un cierto orden que posibilite a los participantes cumplir con los objetivos de entrenamiento planteados en un escenario. Potencialmente se pueden aplicar las técnicas de PM para el análisis y evaluación del entrenamiento de los participantes de un *cyber range*.

Como se visualiza en la Figura 2.2, dada cierta realidad de estudio que tiene asociado un proceso, si se cuenta con algún sistema (generalmente se tiene un sistema de *software*) que recolecta y registra información de la ejecución del proceso, entonces es posible aplicar minería de procesos. Estos registros son primordiales, pues las actividades de PM se basan en estos.

En minería de procesos, el log de eventos (*event log*) recopila los registros de ejecución de un proceso. Este log contiene entradas que, como mínimo, tienen asociada la siguiente información:

- Actividad: corresponde al paso ejecutado del proceso.
- Caso: es la instancia del proceso. Esto quiere decir que todas las actividades con un mismo caso conforman entonces una instancia o una ejecución del proceso.
- *Timestamp*: indica el momento en el que se ejecutó la actividad, lo cual

permite definir el orden de ejecución entre las actividades.

En síntesis, la idea de la minería de procesos es descubrir, monitorear y mejorar procesos reales, extrayendo conocimiento de registros recopilados de la ejecución del proceso [33]. Existen tres grandes actividades que se realizan como parte de la minería de procesos. A continuación, se realiza una breve presentación de cada una de estas.

El descubrimiento de procesos (*process discovery*) implica la generación de modelos de proceso tomando como insumo el log de eventos. Existen múltiples algoritmos de descubrimiento como pueden ser: $\alpha$-*algorithm* [34], *heuristic mining* [35], *inductive mining* [36], entre otros. Adicionalmente, existen múltiples representaciones de modelos, algunos ejemplos son: Petri *nets* [37], Business Process Modeling Notation (BPMN) [38] y *process trees* [39].

La segunda actividad es la verificación de conformidad (*conformance checking*). La idea es que dado un modelo previo del proceso y un log de eventos con registros de ejecución de este mismo proceso, se comparan y detectan posibles desviaciones. Con esta actividad es posible verificar si la realidad registrada en el log de eventos se ajusta al modelo y viceversa. Existen dos grandes algoritmos que permiten detectar, cuantificar y analizar desviaciones, estos son: *token replay* [40] y *alignments* [41].

Por último, se tiene la actividad de mejora de procesos (*process enhancement*). El objetivo en este caso es extender o mejorar un modelo que se tiene previamente construido sobre un proceso, incorporando información de dicho proceso registrada en un log de eventos. Un tipo de mejora es la reparación, donde se modifica el modelo para que represente la realidad de forma más fidedigna en caso de que se identifiquen errores. Otro tipo de mejora es la extensión, donde se incorpora al modelo una nueva perspectiva de análisis que pueda enriquecer el estudio del proceso.

Las perspectivas de análisis hacen referencia a puntos de vista del proceso que resultan interesantes estudiar y por lo tanto se prioriza su análisis. Existen cuatro posibles perspectivas. La perspectiva de flujo de control pone el foco en las actividades y en particular el orden de ejecución. En la perspectiva organizacional, el análisis se centra en los recursos asociados al proceso, como pueden ser los actores y departamentos dentro de una organización que participan de las actividades. Por otra parte, la perspectiva de casos estudia alguna característica o propiedad de los casos (que no implica las anteriores perspectivas). Por último, en la perspectiva de tiempo el foco de estudio es el

tiempo de ejecución de actividades y tiempo total del proceso. Por ejemplo, es muy común estudiar cuellos de botella o actividades del proceso donde la ejecución se torna más lenta e ineficiente.

En definitiva, en nuestro contexto de trabajo pretendemos emplear PM para el estudio de la actividad realizada por los participantes de entrenamientos en un *cyber ranges*. Buscamos descubrir, modelar y analizar los procesos de ataque y defensa ejecutados por los participantes en una infraestructura de entrenamiento, y comparar estos procesos descubiertos con procesos de referencia que detallen la actividad ideal de entrenamiento, de forma que un instructor pueda entender cómo se entrenan los participantes.

## 2.3. *Frameworks* MITRE

MITRE es una organización sin fines de lucro de Estados Unidos que tiene como objetivo el análisis y resolución de problemáticas globales que afectan a la sociedad. Uno de sus campos de trabajo es el área de la ciberseguridad. En este sentido, su objetivo es "*mejorar la capacidad de organizaciones de identificar, detectar, proteger, responder y recuperarse de amenazas y vulnerabilidades*" [42]. Si bien MITRE ha realizado múltiples aportes en el área de ciberseguridad, en nuestro trabajo hacemos uso de los *frameworks* MITRE ATT&CK y MITRE D3FEND.

Estos *frameworks* permiten la categorización de actividad de ataque y defensa en base a tácticas y técnicas. En la Figura 2.3 se puede observar un diagrama de los *frameworks* así como la relación entre estos. Estos *frameworks* representan las tácticas, que son los objetivos de alto nivel, y las técnicas, que detallan como cumplir con dichas tácticas, en formato de matriz. En dicha figura, la primer fila corresponde a las tácticas mientras que las restantes filas representan a las técnicas. Cada táctica tiene múltiples técnicas y una misma técnica puede pertenecer a más de una táctica. Adicionalmente, se encuentra representada la relación existente entre ambos *frameworks*. Las acciones realizadas por un actor malicioso generalmente tienen un impacto en un sistema o una red y generan artefactos digitales (por ejemplo archivos o procesos). Los defensores tratarán de interactuar con dichos artefactos (por ejemplo observándolos o eliminándolos) para responder a las acciones ofensivas.

MITRE ATT&CK es una base de conocimiento que detalla las tácticas y técnicas de los adversarios empleadas en ataques. Dichas tácticas y técnicas

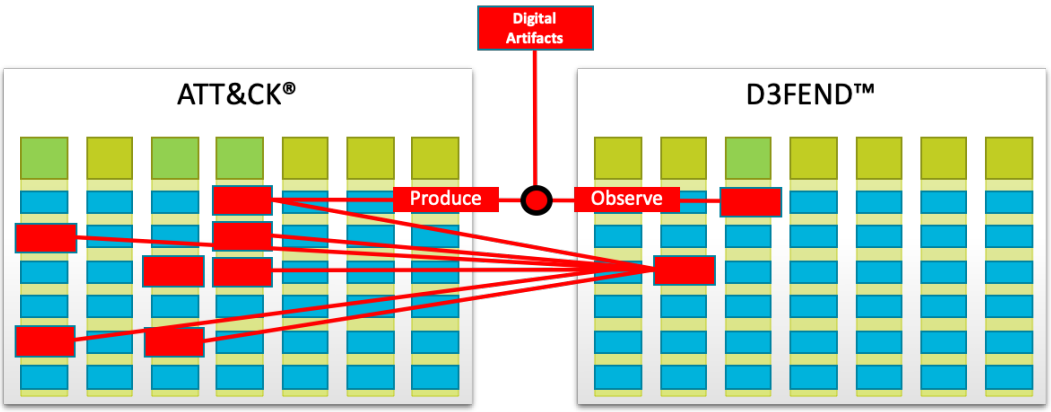



**Figura 2.3:** *Frameworks* MITRE ATT&CK y D3FEND. Extraído de [43].

fueron generadas a partir de observaciones del mundo real, es decir, a partir del estudio de ataques ocurridos contra infraestructuras de organizaciones. Es un recurso fundamental para desarrollar modelos y metodologías de amenazas específicas dentro del sector privado, las agencias gubernamentales y la comunidad de productos y servicios de ciberseguridad [44].

El marco MITRE ATT&CK caracteriza la actividad maliciosa a través de Tacticas, Técnicas y Procedimientos (TTP) adoptados por los adversarios para comprometer un entorno computacional. Las tácticas refieren a los objetivos o metas que los atacantes intentan lograr, mientras que las técnicas y los procedimientos describen cómo logran dichos objetivos con tareas concretas. Adicionalmente, este marco cataloga actores de ciberataques y las herramientas empleadas en estos ataques.

Las TTP suelen ser representadas en una matriz. Existen tres matrices distintas: empresarial (ataques destinados comúnmente a infraestructuras de organizaciones y que implican ataques a sistemas ampliamente utilizados como Windows y Linux), *mobile* (ataques que tienen como objetivos particulares a los dispositivos móviles) e Industrial Control System (ICS) (ataques específicos a infraestructuras asociadas a sistemas de control industrial e infraestructuras críticas). En este trabajo nos enfocamos en el uso de las tácticas y técnicas de la matriz empresarial. Dicha matriz abarca un total de 14 tácticas y 203 técnicas.

Acto seguido se realiza una breve presentación de las tácticas del *framework*:

- Reconocimiento: el adversario intenta recopilar información que pueda

usar para planificar futuros pasos del ataque.

- Desarrollo de recursos: el adversario intenta establecer recursos que pueda usar en el ataque.
- Acceso inicial: el adversario intenta acceder a una red.
- Ejecución: el adversario intenta ejecutar código malicioso.
- Persistencia: el adversario intenta mantener su posición.
- Escalada de privilegios: el adversario intenta obtener permisos o privilegios superiores a los actuales.
- Evasión de defensa: el adversario intenta evitar ser detectado.
- Acceso a credenciales: el adversario intenta obtener nombres de cuenta y contraseñas.
- Descubrimiento: el adversario intenta comprender la infraestructura.
- Movimiento lateral: el adversario intenta obtener acceso a otros sistemas.
- Recopilación: el adversario intenta recopilar datos de interés para su objetivo.
- Comando y Control: el adversario intenta comunicarse con los sistemas comprometidos para controlarlos.
- Exfiltración: el adversario intenta robar datos.
- Impacto: el adversario intenta manipular, interrumpir o destruir sistemas y datos.

Cada táctica puede ser asociada a una fase de un ataque que busca comprometer una infraestructura. Por lo tanto, las tácticas tienen un orden natural de ejecución, que es igual al orden en que fueron presentadas. Este orden surge de las observaciones sobre ataques realizadas por MITRE para diseñar el *framework*. Por ejemplo, antes de poder exfiltrar información es necesario recopilarla, o antes de persistir la posición lograda sobre un sistema es necesario realizar el acceso inicial a este.

Las técnicas permiten explicar en mejor medida cómo hace un adversario para cumplir con una táctica. A modo de ejemplo, algunas de las técnicas asociadas a la táctica de acceso a credenciales son: ataques de fuerza bruta (que implica adivinar credenciales en base a intentos repetidos de autenticación, por ejemplo en una aplicación web, hasta encontrar credenciales válidas) o *network sniffing* (que implica monitorear y analizar el tráfico de red en búsqueda del contenido de paquetes, por ejemplo, para identificar credenciales enviadas por el usuario a la hora de realizar la autenticación en cierta aplicación o sistema).

Por otra parte, el MITRE D3FEND complementa el marco MITRE ATT&CK al centrarse en técnicas y estrategias defensivas. Es una base de conocimientos integral que documenta las técnicas de contramedidas de ciberseguridad. Sirve como un catálogo de estrategias de ciberseguridad defensivas y delinea sus interconexiones con las técnicas ofensivas descritas en MITRE ATT&CK. Al igual que su contraparte, el marco MITRE D3FEND describe las tácticas y técnicas pero que ahora emplean los defensores para proteger una infraestructura. Las tácticas se refieren a las maniobras que ejecutan los defensores en respuesta a un adversario, esencialmente "el qué" de una acción. Por el contrario, las técnicas describen los métodos utilizados para implementar esas tácticas, esencialmente "el cómo" de llevar a cabo la estrategia [45].

Este *framework* cuenta con un total de 7 tácticas y 148 técnicas. A continuación se presentan brevemente las tácticas:

- Modelar: implica aplicar ingeniería de seguridad, análisis de vulnerabilidades, amenazas y riesgos a los sistemas digitales. La idea es identificar y comprender los sistemas que se defienden, las operaciones en dichos sistemas, los actores que los utilizan y las relaciones e interacciones entre estos elementos.
- Endurecer: esta táctica se utiliza para aumentar el coste de oportunidad de la explotación de un sistema o red informática.
- Detectar: se utiliza para identificar el acceso del adversario o la actividad no autorizada en las redes informáticas.
- Aislar: la idea es crear barreras lógicas o físicas en un sistema que reducen las oportunidades de que los adversarios logren más accesos.
- Engañar: esta táctica se utiliza para atraer y permitir a los atacantes potenciales acceder a un entorno controlado para el estudio de sus acciones.
- Expulsar: implica remover a un adversario de una red o sistema informático.
- Restaurar: esta táctica implica devolver el sistema al estado anterior al ser atacado, es decir a su estado de funcionamiento aceptable.

Al igual que MITRE ATT&CK, en el *framework* MITRE D3FEND las tácticas tienen un orden natural de ejecución, que es igual a como fueron presentadas. Por ejemplo, antes de expulsar a un atacante de un sistema es necesario haberlo identificado, y generalmente las tareas asociadas a modelar y endurecer ocurren al inicio de un proceso de defensa.

Cada técnica detallada en este marco de trabajo está asociada a una táctica. A modo de ejemplo, posibles técnicas asociadas a la táctica de expulsión son la revocación de credenciales comprometidas o la inhabilitación de cuentas de usuarios comprometidas con el objetivo que no sean más útiles para el adversario.

Es importante aclarar que mientras MITRE ATT&CK es un proyecto que cuenta con varios años de vida (su primera versión data del año 2018) y ha sido adoptado de forma masiva por la comunidad de ciberseguridad, MITRE D3FEND es un proyecto incipiente. Recién en el año 2024 lanzó su primera versión estable.

La idea de incorporar los *freameworks* MITRE ATT&CK y MITRE D3FEND en nuestro trabajo radica en que permiten reutilizar todo el conocimiento embebido en estos por los expertos en seguridad que los diseñaron, y particularmente permite describir y representar la actividad llevada a cabo por los participantes como parte de los procesos ofensivos y defensivos realizados en los distintos escenarios de entrenamiento de un *cyber range*. De esta forma el entrenamiento de los participantes es expresado en un lenguaje común, facilitando así la comprensión de los instructores sobre las actividades ejecutadas como parte del entrenamiento. Adicionalmente, el uso de estos *frameworks* posibilita analizar la actividad de los participantes en distintos niveles de granularidad, siendo posible estudiar su actividad a niveles de tácticas, que representan los objetivos a alto nivel, o a nivel de técnicas, las cuales permiten entender con mayor detalle cómo cumplen con esos objetivos.

# Capítulo 3

# Tectonic

En este capítulo se presenta Tectonic, una solución de *cyber range* diseñada e implementada desde el GSI. Se introduce un breve contexto de cómo surge el proyecto y se explican las principales funcionalidades y componentes de la plataforma. Adicionalmente, se realiza una comparación con otras soluciones de *cyber range*. Entender aspectos básicos de la plataforma y cómo es su funcionamiento es necesario dado que posteriormente se presentarán casos de estudio que hacen uso de esta plataforma.

## 3.1. Contexto del proyecto

El GSI opera desde el año 2007 el Laboratorio de Seguridad Informática (LaSI) [46]. Este laboratorio fue utilizado para complementar, de forma práctica, el componente teórico brindado en los múltiples cursos de grado y posgrado ofrecidos por el grupo. Sin embargo, en los últimos años surgió la necesidad de mejorar el laboratorio, incorporando tecnologías más actuales y avanzadas. Luego de un proceso de reingeniería llevado a cabo desde 2020, es que surge el *cyber range* académico Tectonic. En particular, nuestro rol dentro de este proyecto ha sido el de analizar, diseñar e implementar varias de las funcionalidades de la plataforma. Algunas de estas funcionalidades claves surgen como aporte del presente trabajo de maestría, las cuales serán presentadas próximamente.

Tectonic es un *cyber range open source* diseñado e implementado por el GSI [4, 6]. Actualmente es utilizado para brindar entrenamientos prácticos como parte de los múltiples cursos dictados por el grupo. Adicionalmente, en el contexto de la Red de Excelencia en Ciberseguridad de Latinoamérica y el

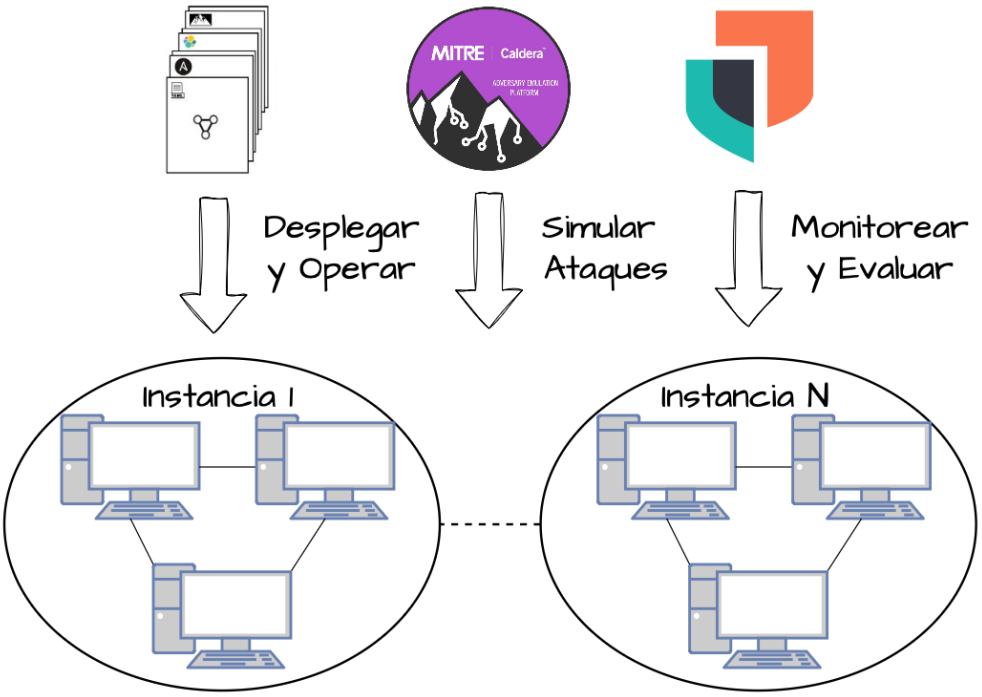



**Figura 3.1:** Funcionalidades claves de Tectonic

Caribe (Red Ciberlac), una red que "*facilita la colaboración y el intercambio de conocimientos, experiencias y recursos entre las instituciones que la integran*" [47], se está impulsando el uso de esta tecnología para que pueda ser empleada por las múltiples universidades de la región.

Este *cyber range* académico está destinado principalmente a docentes, quienes pueden plantear entrenamientos a sus estudiantes de grado y posgrado, de forma que complementan los conceptos teóricos adquiridos, generando y desarrollando habilidades prácticas en la materia.

## 3.2. Principales funcionalidades

Tectonic cuenta con tres funcionalidades claves diagramadas en la Figura 3.1.

En primer lugar, tiene la capacidad de gestionar todo el ciclo de vida de los escenarios de entrenamiento, desde la concepción del escenario, es decir desde su diseño y definición, hasta que efectivamente es desplegado y queda accesible para que los aprendices se entrenen. Para esto, se tiene un lenguaje de especificación de escenarios, el cual permite definir las máquinas que componen el

escenario, cómo se organizan en una topología de red y las configuraciones que se desean aplicar sobre estas. Estas configuraciones pueden ser, por ejemplo, la instalación de software, la creación de usuarios, entre muchas otras. Adicionalmente, la especificación del escenario permite definir configuraciones a ser aplicadas en los servicios de monitoreo en tiempo real y de generación de ataques que incorpora el *cyber range*. Este lenguaje de especificación de escenarios está basado en el lenguaje YAML y *playbooks* Ansible. Lo positivo de contar con este lenguaje es que permite estandarizar la definición del escenario, facilitando el proceso de generación y *testing* de nuevos escenarios. Además, posibilita que los escenarios puedan ser compartidos entre distintas instancias de Tectonic para su despliegue, fomentando así la generación de nuevo contenido que pueda ser utilizado en el *cyber range* por los distintos miembros de la comunidad.

Una vez que se tiene el escenario definido, es posible automatizar el despliegue de los distintos componentes que lo conforman. El despliegue puede ser realizado de forma transparente en la nube de Amazon Web Services (AWS) o en servidores locales utilizando Libvirt o Docker como capa de virtualización. Adicionalmente, Tectonic cuenta con funcionalidades que permiten: destruir un escenario, consultar el estado de un escenario e iniciar o detener máquinas que fueron desplegadas como parte del escenario.

Dado que Tectonic es utilizado principalmente para el entrenamiento de estudiantes en cursos de grado y posgrado, tiene la capacidad de desplegar múltiples instancias de un escenario. La idea es que cada estudiante se entrena en una instancia asignada que se encuentra aislada de las restantes instancias, evitando así que los estudiantes interfieran entre sí durante su entrenamiento. Las instancias son copias de un mismo escenario que pueden ser parametrizadas aplicando cierta configuración única. Generalmente, lo que se individualizan son piezas de información denominadas *flags*. Estas *flags* son esencialmente *strings* que se suelen dejar en archivos y que el estudiante debe identificar y entregar como parte de la resolución del escenario. Tectonic automatiza el despliegue de las *flags*, primero generándolas de forma pseudoaleatoria en base a una semilla y luego desplegándolas a través de *playbooks* Ansible. Como las banderas son generadas a través de una semilla, los docentes pueden obtener los valores originales de las banderas y compararlos con los valores entregados por los estudiantes.

La segunda funcionalidad clave de Tectonic, es la de monitoreo y detección

en tiempo real de actividades realizadas por los aprendices en su entrenamiento. Para esto se hace uso de la tecnología Elastic Security [48], que es un SIEM. Dicha herramienta posibilita la recolección de eventos de la infraestructura de entrenamiento y su análisis para la detección de cierto tipo de comportamiento, generalmente actividad maliciosa. A través de esta funcionalidad el docente puede evaluar el entrenamiento de los estudiantes y brindar *feedback* sobre este. Si bien el monitoreo de la actividad de participantes de entrenamientos empleando el SIEM Elastic Security surge como una idea propuesta en un proyecto de grado de 2021 [49], el diseño e implementación de dicha propuesta como parte de Tectonic es una de las contribuciones realizadas en el presente trabajo de maestría.

Para incluir esta herramienta en Tectonic, primero fue necesario definir donde se desplegaría. En este sentido se definió lo que se llama una red de servicios, una red especial, distinta a las redes de las instancias del escenario, donde se despliegan aquellas máquinas que ofrecen cierto tipo de servicio a dichas instancias. Adicionalmente fue necesario automatizar, a través de *playbooks* Ansible, la instalación de la herramienta Elastic Security así como la instalación de Packetbeat y los agentes de Elastic.

Se definieron dos formas de monitorear el entrenamiento de los estudiantes. Una de estas formas es utilizando exclusivamente Packetbeat, que sirve para la recolección de trazas de red. El punto a favor del uso de este componente es que no es instalado en las máquinas finales sobre las que se entrena el estudiante, por lo cual el estudiante no puede conocer que su actividad está siendo monitoreada. Sin embargo, como solamente se recolectan trazas de red se tiene una menor visibilidad sobre la actividad del estudiante, y en particular para todas aquellas trazas de red que conforman paquetes involucrados en comunicaciones que utilizan protocolos cifrados (como es el caso de Hypertext Transfer Protocol Secure (HTTPS)) no es posible analizar su contenido. En cambio, el otro método de monitoreo implica la instalación del Elastic Agent en las máquinas en las que el estudiante realiza el entrenamiento. Este agente permite una mayor visibilidad sobre la actividad realizada, dado que tiene la capacidad de recolectar: logs, métricas de uso de recursos de *hardware* y otros eventos y trazas de auditoría como pueden ser la ejecución de comandos y procesos, el establecimiento de conexiones de red, la creación de archivos, entre otros. La problemática asociada al uso de este agente es que al ser instalado en las máquinas, el estudiante puede reconocer que está siendo monitoreado e

incluso interferir con el monitoreo (por ejemplo desinstalando el agente).

Si bien al encontrarse en un contexto académico se podría asumir como condición que el estudiante no pude interferir con el agente de Elastic, entendemos que en ciertos casos esto puede limitar o condicionar sus acciones, principalmente en escenarios ofensivos, y tener un impacto negativo en el realismo del escenario y por lo tanto su entrenamiento. A modo de ejemplo, en un escenario ofensivo donde el estudiante debe atacar una infraestructura, puede aplicar una herramienta de ataque que interfiera con las capacidades del agente, incluso sin que el estudiante sea consciente de esto. Es muy común que herramientas de ataque y *malware*, como parte de sus actividades, busquen identificar y deshabilitar herramientas de defensa instaladas en un objetivo bajo ataque. El agente de Elastic, al ser una herramienta de defensa no está exenta de esta realidad. A pesar que el agente tiene la capacidad de protegerse ante manipulaciones maliciosas o indebidas [50], esta funcionalidad está incluida bajo suscripciones pagas de la herramienta a la cual en la actualidad no tenemos acceso. Sin embargo, en una versión del *cyber range* que incluya acceso a este tipo de funcionalidad permitiría el uso del agente de Elastic con una capa extra de seguridad donde no sería posible para el estudiante interferir con la herramienta.

En particular, la instalación de Packetbeat resultó un reto para el caso del despliegue de AWS. Si bien en los casos de Docker y Libvirt basta con desplegar Packetbeat en el mismo *host* donde se realiza el despliegue de los contenedores o máquinas virtuales, ya que de esta forma Packetbeat puede obtener las trazas de red generadas desde dichos orígenes pues tiene acceso a sus interfaces de red, en el caso de AWS esto no es posible. Para AWS fue necesario incluir una nueva máquina en la red de servicios, que cuenta con la instalación de Packetbeat, y a través de la funcionalidad de espejado de tráfico [51] ofrecida por el propio AWS, duplicar el tráfico de red generado por las instancias de entrenamiento para ser reenviado a la máquina que cuenta con la instalación de Packebteat para su monitoreo.

Por otra parte, fue necesario extender el lenguaje de especificación de escenarios para incorporar primitivas que permitieran el uso de Elastic Security. Se incorporaron primitivas que permiten especificar: si es necesario el uso de Elastic Security en un escenario así como los recursos de *hardware* que tendrá asignada la máquina donde se despliega este servicio, el tipo de monitoreo que se quiere realizar ya sea a través de Packetbeat o los agentes de Elastic, y en

este último caso, en qué máquinas del escenario debe ser desplegado el agente. Además, como parte de la especificación del escenario se pueden definir artefactos como: visualizaciones, *dashboards* y reglas de SIEM, que Tectonic desplegará de forma automática (utilizando Ansible) en Elastic Security para su posterior uso por el docente. Las visualizaciones y *dashboard* permiten generar distintos tipos de gráficas para presentar los eventos almacenados en Elastic Security, mientras que las reglas de SIEM permiten correlacionar y analizar los eventos. Más adelante se proporcionará mayor detalle sobre las reglas de SIEM debido a que son un punto fundamental en la metodología de evaluación propuesta.

Por último, la tercera funcionalidad clave es la de ejecución de ataques automatizados sobre la infraestructura de entrenamiento. Esta funcionalidad había sido propuesta como parte de un proyecto de grado realizado en 2023 utilizando la herramienta Caldera [52]. Sin embargo, la integración de la herramienta a Tectonic es otra de las contribuciones realizadas en el presente trabajo de maestría. Caldera es un *framework* y herramienta que permite la definición y emulación de actividad maliciosa sobre una infraestructura [53], basada fuertemente en el *framework* MITRE ATT&CK. Permite replicar las actividades realizadas por actores maliciosos en el contexto de un ataque.

Caldera cuenta con una gama de adversarios, cada uno compuesto por un conjunto de habilidades. Estas habilidades son, en esencia, tareas o actividades que un actor malicioso podría realizar en su ataque. Cabe destacar que estas actividades están meticulosamente catalogadas por el *framework* MITRE ATT&CK. Por ejemplo, algunas actividades se centran en la adquisición de credenciales de usuario, mientras que otras buscan exfiltrar información desde un sistema atacado. Si bien Caldera incluye habilidades y adversarios predefinidos, los docentes pueden diseñar e implementar nuevas habilidades y adversarios para crear ataques contra la infraestructura desplegada por Tectonic.

Para ejecutar un ataque (adversario), se debe definir una operación. La operación implica instanciar al adversario en uno o más agentes de Caldera. Estos agentes son programas instalados en las máquinas que se atacarán o desde donde se lanzará el ataque, y se comunican periódicamente con el servidor de Caldera para recibir instrucciones que ejecutar como parte del ataque. Además, los agentes reportan los resultados de las instrucciones ejecutadas a Caldera, lo que puede ayudar a determinar el flujo de ejecución de un ataque.

Esto significa que, dependiendo del resultado de una habilidad en particular, se puede decidir la habilidad que se ejecutará posteriormente, lo que permite ataques más complejos y realistas.

Existen varios tipos de agentes de Caldera. Se incorporó a Tectonic una funcionalidad que permite la instalación automatizada de agentes que se comunican con Caldera mediante el protocolo Hypertext Transfer Protocol (HTTP). El docente puede especificar las máquinas en las que se instalarán estos agentes como parte de la definición del escenario. Al tener un agente instalado en una máquina que ejecuta actividad maliciosa, se está asumiendo que la fase inicial de compromiso del ataque ya fue exitosa. Sin embargo, si el docente desea simular el acceso inicial a la máquina, puede utilizar el agente instalado en el servidor de Caldera para obtener acceso inicial a la máquina atacada.

En resumen, la integración de Caldera en Tectonic brinda la oportunidad de simular ataques realistas e implementar escenarios defensivos. Estos escenarios se pueden clasificar en dos tipos: *offline* y *online*. En los escenarios *offline* (o *post-mortem*), se puede utilizar Caldera para lanzar ataques a la infraestructura y generar logs y artefactos que permitan a los estudiantes practicar habilidades y usar herramientas del ámbito de la forensia digital [54]. Por otro lado, en escenarios *online*, Caldera puede orquestar ataques en tiempo real donde los estudiantes deben tomar medidas defensivas, como detectar intrusiones y recuperar sistemas para responder ante estos ataques.

Para la integración de Caldera con Tectonic fue necesario realizar ciertas modificaciones en el *cyber range*. En primer lugar, se desarrollaron *playbooks* Ansible que permiten la instalación y configuración automática de Caldera como una máquina que es desplegada como parte de la red de servicios. Adicionalmente se desarrollaron *playbooks* Ansible para la instalación de los agentes de Caldera en las instancias de entrenamiento. Por otra parte, se extendió el lenguaje de especificación de escenarios para incluir primitivas asociadas a Caldera. Se incluyeron primitivas que permiten indicar: si es necesario el uso de Caldera para un escenario y los recursos de *hardware* asociados a la máquina donde se despliega este servicio, las máquinas del escenario que requieren tener la instalación del agente de Caldera, y los distintos artefactos como habilidades, adversarios y operaciones que Tectonic configurará de forma automática, mediante Ansible, en Caldera y estarán disponible para el uso del docente.

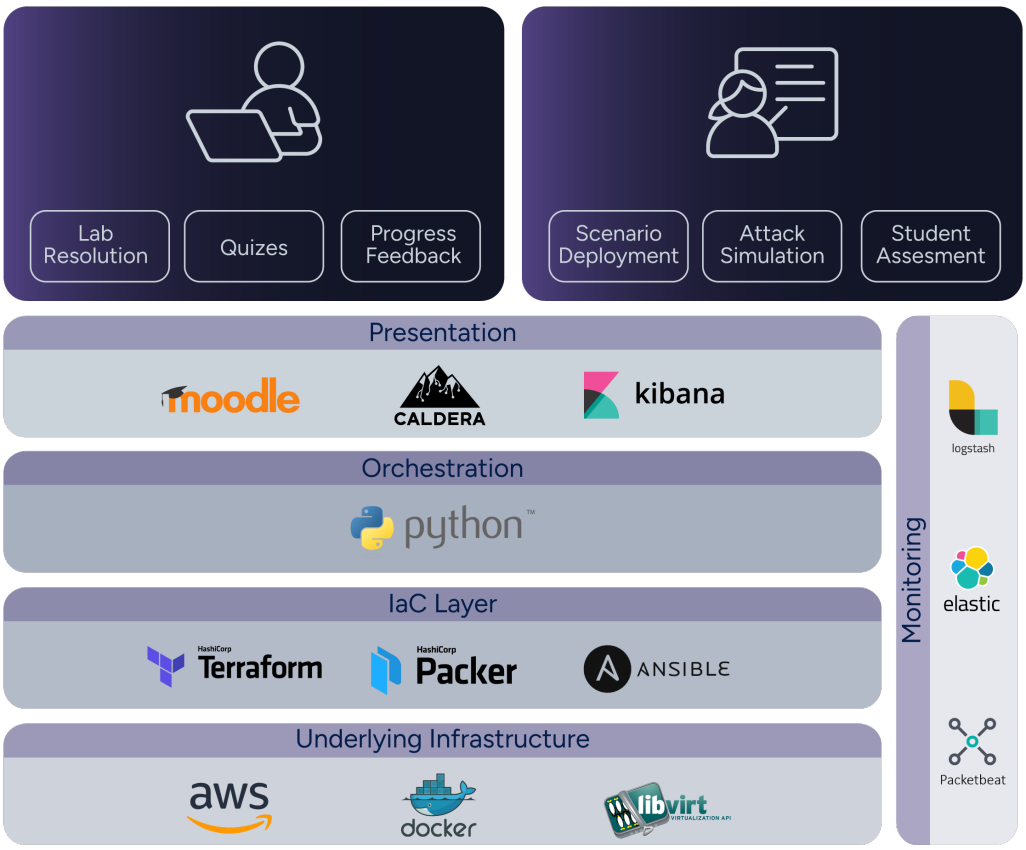



**Figura 3.2:** Arquitectura de Tectonic

## 3.3. Arquitectura

La arquitectura de Tectonic consta de cinco capas fundamentales representadas en la Figura 3.2. A continuación se detalla brevemente cada una de estas capas.

La capa de infraestructura base es donde se realiza el despliegue de los escenarios de entrenamiento. Como fue mencionado, Tectonic permite el despliegue de forma transparente en la nube de AWS o en servidores locales utilizando Libvirt o Docker. Adicionalmente, Tectonic fue diseñado e implementado de forma tal que puede ser fácilmente extendido para incorporar otras tecnologías de despliegue como puede ser otras nubes públicas, por ejemplo Azure [55] o Google Cloud Platform (GCP) [56].

Para el despliegue de escenarios, se hace uso de una capa compuesta por herramientas de Infrastructure as Code (IaC), donde la idea es especificar la infraestructura a desplegar como código lo cual facilita que el despliegue pueda

ser automatizado y repetible [57]. Se utilizan las herramientas: Packer para la generación de imágenes base, Terraform [58] para el despliegue de máquinas y redes, y Ansible para las configuración de las máquinas.

Sobre las anteriores capas se tiene una capa de orquestación. Esencialmente es una aplicación escrita en el lenguaje de programación Python [59] que se encarga de gestionar el despliegue del escenario invocando a los componentes correspondientes.

Por otra parte, la capa de monitoreo es una capa transversal que permite la recolección de eventos desde la infraestructura de entrenamiento. Como base se utiliza la tecnología Elastic Stack [60]. Este *stack* está compuesto de cuatro proyectos *open source*: Elasticsearch, Logstash, Kibana y Elastic Agent o Beats. Elasticsearch es una base de datos no relacional. Logstash es utilizado para la transformación de eventos previo a su almacenamiento. Kibana permite la realización de búsquedas y visualizaciones sobre los datos. Como ya fue presentado, Elastic Agent y Beats (como es el caso de Packetbeat) permiten la recolección de distintos tipos de eventos como pueden ser logs de servidores, trazas de red, métricas de uso de recursos, entre otros. Sobre este *stack* tecnológico se implementan distintos casos de uso. En particular, Tectonic emplea el caso de uso de seguridad, es decir, el SIEM Elastic Security, que como fue mencionado anteriormente, permite al docente la evaluación del entrenamiento de los estudiantes.

Por último, la capa de presentación es utilizada como *frontend* para los usuarios, en particular para el usuario docente. Esta capa está compuesta por Moodle, un LMS que permite gestionar los componentes de carácter más teórico de los entrenamientos como pueden ser cuestionarios; Caldera que como fue presentado anteriormente es utilizado para la implementación de ataques sobre la infraestructura de entrenamiento; y Kibana que permite la visualización y realizar consultas sobre los eventos recolectados de la infraestructura de entrenamiento.

En la actualidad, el uso del *cyber range* por parte de los usuarios docentes es esencialmente a través de una CLI en lo que tiene que ver con la definición de escenarios y su despliegue, e interfaces web de tipo GUI para Caldera y Kibana. De forma similar, el acceso al *cyber range* por parte de los estudiantes también es a través de una CLI dado que utilizan el protocolo SSH para la conexión remota a las máquinas de entrenamiento. En un futuro se tiene pensado mejorar las interfaces de ambos usuarios, para lograr interfaces de tipo GUI

| *Cyber Range* | Público Objetivo | Especificación de escenarios | Despliegue escenarios | Simulación de ataques | Monitoreo y evaluación | Interfaces |
|---|---|---|---|---|---|---|
| Tectonic | Estudiantes universitarios | Lenguaje propio basado en YAML y Ansible | Utiliza AWS, Docker o Libvirt | Lo permite a través de la herramienta Caldera | Basado en cuestionarios, entrega de banderas y monitoreo en tiempo real de la infraestructura | Principalmente de tipo CLI |
| Clusus | Estudiantes universitarios | No fue posible obtener detalles | Utiliza Docker y Virtualbox | No lo permite | Basado en cuestionarios, entrega de banderas y monitoreo en tiempo real de la infraestructura | GUI |
| Cyberwiser.eu | Organizaciones y profesionales | No fue posible obtener detalles | Utiliza AWS, Openstack o OpenNebula | Lo permite pero se desconocen detalles de implementación | Basado en monitoreo en tiempo real de la infraestructura y análisis de vulnerabilidades | GUI |
| CyTrONE | Estudiantes universitarios | Lenguaje propio basado en YAML y script en distintos lenguajes de programación | Utiliza AWS o Libvirt | Lo permite a través de scripts | Basado en cuestionarios y entrega de banderas | GUI |
| KYPO | Estudiantes universitarios | Lenguaje propio basado en YAML y Ansible | Utiliza Openstack | No lo permite | Basado en cuestionarios, entrega de banderas y monitoreo en tiempo real de la infraestructura | GUI |
| THREAT-ARREST | Organizaciones y profesionales | Lenguanje propio que permite definir programas de entrenamiento | No fue posible obtener detalles | Lo permite pero se desconocen detalles de implementación | Basado en monitoreo en tiempo real de la infraestructura y análisis de vulnerabilidades | GUI |

**Tabla 3.1:** Comparativa de *cyber ranges*

que resulten más amigables, intuitivas y en definitiva más sencillas de utilizar.

## 3.4. Comparativa con otras soluciones de *cyber range*

En la actualidad existen múltiples exponentes de plataformas de *cyber range*. En la sección 2.1.3, se presentaron algunas de estas plataformas. Puede resultar interesante comparar Tectonic con otros *cyber ranges* para identificar puntos en común y funcionalidades a incorporar o mejorar. La Tabla 3.1 resume las principales características de cada plataforma según los distintos puntos analizados.

Comenzando el análisis por el público objetivo, se identificó que los *cyber ranges* Clusus, CyTrONE y KYPO tienen como público objetivo estudiantes universitarios, mientras que THREAT-ARREST y Cyberwiser.eu tienen como foco organizaciones y profesionales. Si bien hasta el momento el público objetivo de Tectonic se ha enfocado en estudiantes universitarios, dado que ha sido empleado, principalmente, para brindar entrenamiento práctico en cursos de grado y posgrado ofrecidos por el GSI, también es posible plantear escenarios de entrenamientos destinados a profesionales y equipos de seguridad de organizaciones y empresas.

En relación a lenguajes de especificación de escenarios, cada *cyber range*

implementa su propio lenguaje. Si bien en términos generales todos permiten la especificación de máquinas a desplegar, la topología de red asociada y las configuraciones a aplicar sobre las máquinas, las herramientas que se utilizan suelen variar. En el caso de CyTrONE se identificó que utiliza herramientas más básicas como *scripts* (que pueden ser implementados en distintos lenguajes como Bash o Python) para la configuración de las máquinas. En cambio, casos como los de KYPO y Tectonic hacen uso de Ansible para la especificación de las configuraciones, lo cual facilita en gran medida la definición y aplicación de las tareas a aplicar sobre las máquinas ya que es una muy buena herramienta de gestión y automatización de configuraciones, fácil de utilizar y que cuenta con una fuerte comunidad que brinda soporte a la herramienta.

En lo que refiere a funcionalidades, y en particular en la automatización del despliegue de escenarios, Tectonic permite el despliegue de forma transparente sobre distintas infraestructuras bases, tanto en la nube como de forma local, y puede ser fácilmente extendido para integrarse con nuevas tecnologías de despliegue. En términos generales, las restantes plataformas también permiten el despliegue de escenarios tanto de forma local como en nubes públicas utilizando distintas herramientas de virtualización.

La posibilidad de simular ataques automáticos sobre la infraestructura de entrenamiento es una funcionalidad vital que Tectonic ofrece para la implementación de escenarios defensivos. Plataformas como CyTrONE y CyberWiser.eu también ofrecen esta funcionalidad, mientras que otros *cyber ranges* como KYPO o Clusus no parecen ofrecerla.

En lo que refiere al monitoreo y seguimiento del entrenamiento de usuarios, la gran mayoría de *cyber ranges* ofrecen funcionalidades para dicha tarea. Los casos de KYPO, Clusus y CyberWiser.eu son bastante similares al de Tectonic ya que permiten recolectar eventos desde las máquinas donde se entrenan los usuarios para posteriormente ser analizados de forma centralizada. En cambio, casos como los de CyTrONE basan la evaluación exclusivamente en cuestionarios y artefactos entregados por los usuarios a través de la plataforma Moodle. Es importante aclarar que Tectonic también permite este tipo de evaluación pues tiene la posibilidad de integrarse con dicho LMS.

Por último, en cuanto a interfaces presentadas a los usuarios, Tectonic cuenta únicamente con interfaces de tipo CLI. Generalmente, otras soluciones de *cyber ranges* suelen proporcionar interfaces GUI que pueden resultar más accesibles y sencillas de utilizar para los usuarios. Tectonic es un proyecto que

se encuentra en constante evolución, la mejora de las interfaces presentadas a los distintos usuarios se encuentra dentro del trabajo a futuro del proyecto.

## 3.5. Operativa de Tectonic

Cuando un docente desea brindar un entrenamiento a sus estudiantes empleando Tectonic, despliega un escenario. Por lo general, un escenario busca aplicar o tratar cierto concepto teórico en un caso práctico concreto. Como parte del escenario, al estudiante se le plantea una realidad o contexto de trabajo, objetivos de entrenamiento específicos que deberá cumplir y se le disponibiliza una infraestructura que estará conformada por máquinas, redes y configuraciones de dichas máquinas. A modo de ejemplo, un escenario puede tratar de tipos de ataques web, para lo cual puede ser desplegada una máquina víctima que cuenta con una instalación de una aplicación web vulnerable y una máquina atacante desde la cual el estudiante lanzará los ataques. Esta máquina atacante seguramente tenga instaladas distintas herramientas de ataque.

Al desplegar el escenario, realmente lo que se despliega es una cierta cantidad de instancias del escenario. Cada una de estas instancias es una copia aislada del escenario asignada a un estudiante, de forma que cada estudiante se entrena en su propia copia sin influir en el entrenamiento de otros estudiantes. En la práctica lo que se suele hacer para fomentar un ambiente colaborativo es generar grupos de estudiantes, de forma que cada grupo de estudiantes interactúa exclusivamente con su instancia correspondiente. Los escenarios pueden estar disponibles durante el tiempo que el docente lo desee. Se han realizado talleres de entrenamiento que duraron unas pocas horas (ejemplos de estos talleres son presentados en el Capítulo 5) y en otros casos se han dictado cursos cuyos laboratorios prácticos se prolongan durante varias semanas.

Como fue mencionado, las instancias son copias idénticas salvo por el proceso de parametrización aplicado por Tectonic. Este proceso implica que ciertas piezas de información se hacen únicas o individualizan para cada instancia. Estas piezas de información son las *flags* que el estudiante debe identificar y entregar como parte de la resolución del escenario. A modo de ejemplo, y continuando con el escenario del ataque web, si una parte de dicho escenario implica un ataque de tipo Structured Query Language injection (SQLi)[1], entonces una

[1]En este tipo de ataques, el actor malicioso realiza consultas SQL especialmente di-

*flag* podría ser ubicada directamente en una tabla de la base de datos utilizada por la aplicación. El uso de *flags* únicas por cada instancia mitiga o previene casos de copia (reutilización de banderas) entre los propios estudiantes.

Los escenarios suelen ser diseñados como ejercicios de una competencia de tipo CTF [61], de forma tal que si el estudiante identifica la *flag*, entonces es porque ha completado el escenario. La *flag* debe ser entregada a través de la plataforma Moodle, donde adicionalmente al estudiante se le puede solicitar que complete cuestionarios asociados al escenario (estos cuestionarios son individuales, por más que el entrenamiento es realizado en grupos). Una vez finalizado el entrenamiento, el docente comprueba que las *flags* entregadas son correctas para cada estudiante (o grupo) y valida además los resultados de los cuestionarios.

El problema de este enfoque es que la evaluación realizada es esencialmente binaria y restringida. Es decir, que el docente puede evaluar si el estudiante entrega la *flag* lo que significa que ha completado de forma satisfactoria el escenario. No es posible que el docente evalúe cómo se realiza el entrenamiento, qué técnicas se aplican o qué herramientas se emplean.

Como parte de este trabajo de maestría se incorporó a Tectonic la tecnología de SIEM Elastic Security. El uso de esta tecnología posibilita la recolección de eventos desde las máquinas donde se entrenan los usuarios para su posterior análisis y correlación en base a reglas que buscan identificar cierto tipo de actividad o comportamiento. Estas reglas generan alertas, las cuales notifican de la ocurrencia de cierta actividad y que pueden ser analizadas por los docentes. De esta forma, el docente sí puede comprender cómo realiza el entrenamiento el estudiante, ya que es posible identificar técnicas y herramientas aplicadas. Adicionalmente, es importante destacar que Elastic Security realiza un análisis en vivo, es decir, a medida que los estudiantes se entrenan y generan eventos, el SIEM aplica reglas y genera alertas. Por lo tanto, el docente puede ir analizando resultados parciales y en tiempo real a medida que los estudiantes avanzan en el entrenamiento.

Si bien la herramienta Elastic Security es muy potente y brinda completa visibilidad sobre el entrenamiento de los estudiantes en Tectonic, puede ser compleja de utilizar para docentes que no cuentan con experiencia sobre la herramienta ya que la curva de aprendizaje puede ser pronunciada. Además,

señadas para obtener información desde la base de datos que no debería ser accesible a un usuario o incluso alterar o destruir datos de la base de datos.

tiene la desventaja que las alertas generadas no necesariamente tienen una relación fácilmente identificable por el docente, por lo cual puede no ser clara toda la actividad desarrollada por el estudiante. Por ejemplo, si como parte del escenario el estudiante debe atacar una infraestructura, lo más probable es que con Elastic Security se logren identificar ciertas actividades o etapas llevadas a cabo como parte del ataque, pero no necesariamente se logre un entendimiento del ataque completo. Adicionalmente, por ser una herramienta de SIEM, su objetivo primordial es la detección e identificación de actividad maliciosa. Esto dificulta la evaluación de escenarios defensivos donde los participantes toman el rol del defensor, y por lo tanto, el interés radica en estudiar y analizar las acciones de defensa que estos aplican. Por esta razón, surge la necesidad de diseñar una metodología que permita una evaluación íntegra del entrenamiento de los estudiantes, tanto de escenarios ofensivos como defensivos, y que pueda ser aplicada a Tectonic.

# Capítulo 4

# Metodología de Evaluación

En este capítulo se presenta la metodología diseñada para lograr la evaluación del entrenamiento de participantes en un *cyber range*. Un avance de la metodología fue publicado en la conferencia CLEI 2024 [5] y un artículo adicional se encuentra pendiente a ser publicado en el CLEI Electronic Journal [7].

## 4.1. Propuesta de metodología

A la hora de realizar el entrenamiento, el participante es libre de ejecutar aquellas tareas y aplicar las herramientas que considere necesarias para cumplir con los objetivos de entrenamiento a su propio ritmo. Es decir, que si bien el escenario tiene objetivos de entrenamiento definidos, la forma en que el participante alcanza estos objetivos no necesariamente está predefinida dado que pueden existir varias formas de resolución. La existencia de múltiples formas de resolución puede ser tanto de forma consciente, es decir que el instructor diseña el escenario para que existan distintas formas de resolución, o inconsciente, donde el instructor no conoce estos múltiples caminos. Esto último, suele ocurrir debido al uso de sistemas y tecnologías que a medida que pasan los años van presentando nuevas vulnerabilidades de seguridad. Si estas vulnerabilidades están presentes en la infraestructura de entrenamiento, los participantes pueden identificarlas y explotarlas para cumplir con los objetivos de forma alternativa. Por ejemplo, en un escenario ofensivo el objetivo puede ser obtener acceso de *root* (máximos privilegios en sistemas Linux) en una máquina. El instructor puede haber diseñado el escenario para que la máquina

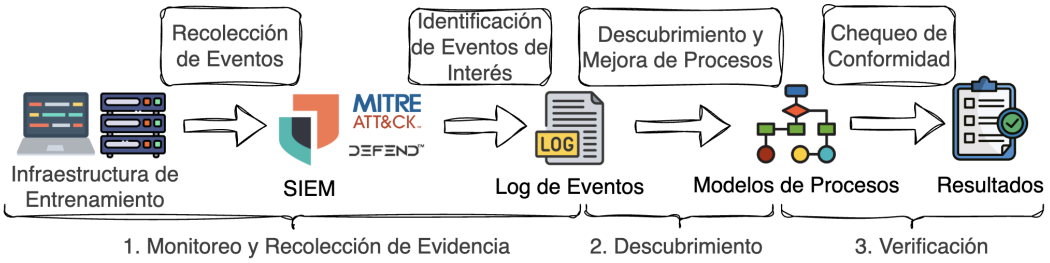



**Figura 4.1:** Metodología de evaluación de entrenamientos en un *cyber range*

presente configuraciones erróneas que le permiten al participante realizar un ataque de escalada de privilegios, es decir que a partir de un acceso inicial con privilegios distinto de *root* utiliza ciertas herramientas y técnicas que le permiten obtener privilegios mayores, en este caso de *root*. Sin embargo, con el paso del tiempo es muy común que los sistemas Linux presenten nuevas vulnerabilidades que posibiliten este tipo de ataques. En caso que el instructor no mantenga actualizado los sistemas utilizados en el escenario, dichas vulnerabilidades estarán disponibles para que el participante pueda explotarlas y cumplir con los objetivos de forma alternativa al diseño planteado por el instructor.

En síntesis, el entrenamiento del participante puede ser entendido como un proceso, ya que realiza distintas actividades o acciones, siguiendo cierto orden, para cumplir con el objetivo de entrenamiento concreto planteado en el escenario. Por tal razón, es que la metodología de evaluación propuesta tiene como pilar fundamental la minería de procesos. Adicionalmente, utiliza otras herramientas y tecnologías como los *frameworks* MITRE ATT&CK y MITRE D3FEND, y un SIEM para el análisis y correlación de eventos.

La Figura 4.1 diagrama la metodología diseñada para la evaluación del entrenamiento en un *cyber range*. Esta metodología consta de tres etapas y brinda una evaluación íntegra del entrenamiento de los participantes, permitiendo al instructor entender no solamente si el participante completa los objetivos de entrenamiento, sino que, más importante aún, cómo lo hace. Además, puede ser utilizada para la detección de trampa por parte de los participantes y el análisis del diseño del escenario.

La metodología permite entender la actividad desarrollada por los participantes en un entrenamiento, identificando tareas particulares, su relación y contextualizandolas como parte de un proceso de ataque o defensa de una infraestructura. Las actividades son expresadas y comunicadas de una forma

estandarizada y en distintos niveles de abstracción mediante el uso de los *frameworks* MITRE ATT&CK y MITRE D3FEND. En las siguientes secciones se presentan cada una de las etapas de la metodología.

### 4.1.1. Monitoreo y recolección de evidencias

La etapa inicial consiste en el monitoreo y recolección de eventos generados por los participantes en la infraestructura de entrenamiento. El artefacto generado al finalizar esta etapa es el log de eventos, utilizado en etapas posteriores para la aplicación de tareas de PM.

Para cumplir con esta tarea se propone el uso de una herramienta de SIEM. Esta tecnología permite la recolección de distintos eventos, generalmente a través de la instalación de un agente en las máquinas que conforman el escenario. Los eventos que pueden ser recolectados son: logs del sistema y distintos servicios, métricas de uso de recursos de *hardware* y otros tipos de eventos de seguridad y auditoría como la ejecución de procesos, el establecimiento de conexiones de red, la creación o lectura de archivos, entre otros.

Los eventos recolectados dependerán en gran medida de la problemática tratada en el escenario. A modo de ejemplo, si el escenario implica un ataque a una aplicación web, seguramente los logs de las transacciones HTTP registradas por el servidor web resulten los más útiles para comprender la actividad del participante. En otros casos, puede resultar de mayor utilidad los comandos y procesos ejecutados por el participante, como en el ejemplo planteado anteriormente donde el participante debe realizar un ataque de escalada de privilegios para obtener acceso *root* a una máquina, dado que las acciones ejecutadas para dicho ataque son visibles únicamente dentro de la máquina atacada. La recolección de los eventos y posterior envío al SIEM se realiza en tiempo real, es decir, a medida que el participante avance en el entrenamiento.

Para la aplicación de tareas de PM, es vital contar con un log de eventos. Este log de eventos, en su forma más completa, podría ser construido a partir de todos los eventos recolectados de la infraestructura de entrenamiento. Sin embargo, no necesariamente todos los eventos recolectados resultan útiles para entender las actividades llevadas a cabo por los participantes. Por ejemplo, en un Sistema Operativo (SO) existen acciones realizadas por el propio sistema para su funcionamiento que no están asociadas o no fueron generadas por el usuario que lo utiliza. Estas acciones no son de interés para la evaluación

del instructor. En otras palabras, el log de eventos debería incluir solamente aquellos eventos significativos que permiten entender las acciones llevadas a cabo por los participantes como parte de la resolución del escenario de entrenamiento. Adicionalmente, como el número de estos eventos es menor que el número de la totalidad de posibles eventos, es esperable entonces que el log de eventos se mantenga acotado en cuanto a su tamaño y esto aportaría a que las siguientes tareas de PM no requieran de recursos de *hardware* significativos pues se trabaja con logs de eventos relativamente pequeños.

Para la detección de los eventos relevantes, también llamados eventos de interés, se utiliza el motor de correlación y análisis que incluye el SIEM. Este motor ejecuta de forma periódica, a medida que los eventos son recolectados, reglas o analíticas de seguridad. Estas reglas son esencialmente consultas sobre los datos que permiten identificar cierto tipo de comportamiento o actividad, correlacionando distintos tipos de eventos. Este es el poder del SIEM, es decir, que brinda completa visibilidad sobre los distintos sistemas y tecnologías que conforman un escenario de entrenamiento, permitiendo un análisis más profundo y preciso.

Las actividades identificadas y detectadas dependerán en gran medida del escenario en cuestión. Por lo tanto, las reglas utilizadas deberán ser ajustadas para cada escenario. En el caso del escenario de escalada de privilegios, se podrían utilizar reglas que identifiquen el acceso al sistema con el usuario sin privilegios, y luego con el usuario con privilegios de *root*. En cambio en un escenario de ataque a una aplicación web se podrían utilizar reglas que identifiquen ciertos tipos de ataques web como puede ser un ataque de SQLi que permitiría un acceso indebido a la base de datos utilizada por las aplicación web. En este sentido, como parte de la especificación del escenario, el instructor debe proveer el conjunto de reglas a utilizar en el SIEM. Asimismo, el instructor debe asegurar que las reglas se encuentren etiquetadas en base a las tácticas y técnicas de los *frameworks* MITRE ATT&CK y MITRE D3FEND. Más adelante, se detallará cómo se utiliza cada uno de estos *frameworks* en concreto.

En consecuencia, el uso de estas reglas de seguridad en el SIEM brinda valor agregado, ya que permite identificar eventos de interés y contextualizarlos, entendiendo acciones ofensivas y defensivas, tareas asociadas y la motivación para ejecutarlas según el conocimiento embebido en los *frameworks* de MITRE.

Una vez que se tienen definidos los eventos de interés se construye el log de eventos. La Tabla 4.1 ejemplifica un posible log de eventos. Existen tres campos

| ***Timestamp*** | **Caso** | **Actividad** |
|---|---|---|
| 2025-02-04T09:00:00 | 1 | Acceso inicial |
| 2025-02-04T09:05:00 | 1 | Descubrimiento |
| 2025-02-06T17:09:00 | 2 | Acceso inicial |
| 2025-02-04T09:21:00 | 1 | Recopilación |
| ... | ... | ... |
| 2025-02-07T11:37:00 | 2 | Descubrimiento |

**Tabla 4.1:** Log de eventos de ejemplo

fundamentales que son necesarios tener para cada evento que conforma este log. El *timestamp* indica el momento en el que ocurrió el evento, el identificador de caso indica el participante que ejecutó la acción; y por último, la actividad es la alertada por la regla del SIEM y representa la acción llevada a cabo por el participante en su entrenamiento. Esta actividad se expresa en términos de tácticas y técnicas de los *frameworks* MITRE ATT&CK o MITRE D3FEND.

En definitiva, el log de eventos recopila la actividad clave realizada por cada participante como parte de su entrenamiento, y que permitirá al instructor entender cómo es que el participante se entrena.

Es importante aclarar que si bien se mencionó que el log de eventos debería incluir solamente aquellos eventos significativos que permiten entender las acciones llevadas a cabo por los participantes como parte de la resolución del escenario de entrenamiento, esto puede tornarse en una tarea muy ardua. Dado que la identificación de los eventos de interés se realiza en base a las reglas de SIEM, es preciso utilizar un conjunto de reglas completo que permita la correcta identificación de dichos eventos. Sin embargo, las reglas de SIEM pueden presentan el problema de falsos positivos y falsos negativos, en otras palabras las reglas pueden presentar errores. A modo de ejemplo, en un escenario ofensivo las reglas utilizadas pueden detectar e identificar actividad maliciosa por parte de los participantes cuando realmente dicha actividad no lo es, o directamente no alertar la ocurrencia de actividad maliciosa cuando ocurre realmente. En consecuencia, estos errores serán propagados al log de eventos y sucesivas etapas de la metodología.

Por lo tanto, las reglas de SIEM utilizadas en cada escenario son un punto fundamental de esta metodología y es altamente deseable que sean cuidadosamente diseñadas, implementadas y probadas. Como se discutirá más adelante, el instructor puede optar por generar sus propias reglas o reutilizar reglas generadas por expertos en ciberseguridad las cuales podrían presentar un mayor

nivel de madurez y confianza en su desempeño.

### 4.1.2. Descubrimiento

La siguiente etapa de la metodología implica la generación de modelos que representan el proceso ejecutado por los participantes para cumplir con los objetivos de entrenamiento planteados en un escenario. Se aplican algoritmos de descubrimiento de procesos para tal fin.

En este trabajo se hace foco en el algoritmo Inductive Miner [36]. Este algoritmo funciona de la siguiente forma: identifica el corte o partición más representativa en el log de eventos y le asigna un operador apropiado. El operador puede ser secuencial (la actividad B ocurre inmediatamente después de la actividad A), paralelo (actividades A y B pueden ocurrir al mismo tiempo) o de elección entre actividades (luego de la actividad A se puede elegir si ejecutar la actividad B o C). El algoritmo continúa recursivamente sobre cada sublog generado debido a la partición realizada. El resultado de la aplicación del algoritmo es un modelo representado por un árbol de procesos que asegura la propiedad de *soudness*. Esta propiedad implica la corrección de los modelos en función de tres características esenciales: asegurar que las trazas finalizan, no existen *deadlocks* o *loops* infinitos que impidan que una traza avance en su ejecución y asegurar la ausencia de actividades que no pueden ser ejecutadas [62]. En definitiva, el uso de este algoritmo garantiza la generación de modelos correctos según estas propiedades, lo cual es crucial para los análisis posteriores que se puedan realizar sobre ellos. Los modelos que no cumplen estas propiedades pueden ser difíciles de interpretar y analizar, lo que podría dar lugar a resultados erróneos.

Para la aplicación de PM se hace uso de distintas herramientas. En el mercado existen múltiples herramientas de PM como pueden ser ProM Tools [63, 64], Disco [65, 66], PM4PY [67, 68] o Breamline [69]. En este trabajo se han utilizado las herramientas Disco para el filtrado y análisis inicial del log de eventos, y ProM Tools para la aplicación de algoritmos de descubrimiento. La razón del uso de estas herramientas radica en su disponibilidad de forma gratuita o a través de licencias académicas, su facilidad de uso y funcionalidades brindadas. En particular, para el descubrimiento de procesos se utiliza el *framework* Inductive Miner [70, 71] incluido como un *plugin* en la herramienta ProM Tools. Sin embargo, otras herramientas pueden ser utilizadas para la

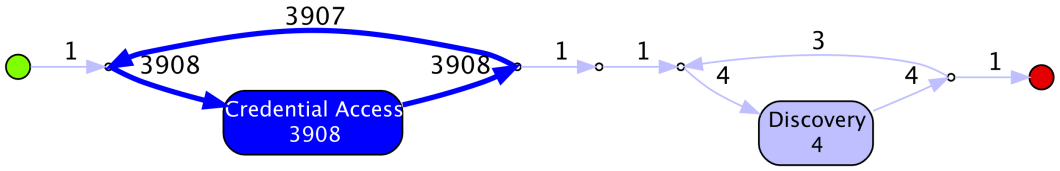



**Figura 4.2:** Ejemplo de modelo obtenido a raíz de la aplicación de descubrimiento de procesos.

implementación de las tareas de minería de procesos.

En el contexto de esta metodología de evaluación, la tarea de descubrimiento de procesos puede ser aplicada con dos fines distintos. En primer lugar, el objetivo más directo es el descubrimiento de modelos que representan las actividades llevadas a cabo por los participantes como parte de su entrenamiento. Estos modelos visuales pueden ser analizados por los instructores para entender cómo ha ido avanzando el participante en su entrenamiento. En base a este análisis, el instructor puede brindar *feedback* al participante sobre su rendimiento para que pueda mejorar sus habilidades.

En la Figura 4.2 se ejemplifica un modelo obtenido luego de la aplicación de descubrimiento de procesos. El modelo representa el punto de inicio (punto verde), el punto final (punto rojo) y todas las actividades ejecutadas en el entrenamiento. Los números asociados a las actividades y rutas indican la cantidad de ejecuciones, y el color azul más oscuro representa las actividades y rutas que se ejecutan una mayor cantidad de veces. Este modelo podría representar la actividad llevada a cabo por un participante en un escenario ofensivo donde primero se realizan ataques que le permiten acceder a credenciales de cuentas de usuario de una máquina, y posteriormente, con acceso a la máquina, ejecutar tareas que le permiten reconocer y obtener información sobre la máquina (por ejemplo: cuentas de usuarios, herramientas y aplicaciones instaladas, procesos en ejecución, entre otros) para desarrollar futuros ataques. En el modelo se identifica claramente que las acciones de obtención de credenciales ocurren con anterioridad a las acciones de descubrimiento, lo cual tiene sentido ya que para ejecutar acciones de descubrimiento primero se debe contar con acceso a la máquina atacada, es decir se debe contar con las credenciales de acceso. Adicionalmente se identifica una predominancia de las tareas de acceso a credenciales que son ejecutadas en *loop* una gran cantidad de veces.

En cambio, la segunda aplicación puede ser con el fin de generar modelos de referencia que de cierta forma representan la actividad ideal que debe realizar

el participante para cumplir con los objetivos de un escenario. Estos modelos son utilizados en la siguiente etapa de la metodología para la comparación y verificación de resultados. Es importante destacar que a medida que ocurren las sesiones de entrenamiento y se recopilan nuevos logs de eventos (de un mismo escenario), el modelo de referencia del escenario puede ser mejorado o extendido aplicando la técnica de reparación asociada a la mejora de procesos. La idea es que al obtener nuevos registros de la ejecución del proceso, se podrían identificar nuevas actividades que anteriormente no fueron registradas, y por lo tanto resultaría de utilidad reparar el modelo de referencia para que también logre incorporar y representar estas nuevas actividades.

Por último, el instructor que diseña un escenario tiene un gran conocimiento sobre los objetivos de entrenamiento y pudo haber identificado tareas y actividades necesarias para cumplir con dichos objetivos, así como la relación entre dichas actividades. Por esta razón, el instructor podría utilizar este conocimiento para generar modelos de referencia de forma manual que sirvan como una primera aproximación a la actividad que se espera que ejecute el participante para cumplir con el entrenamiento. Adicionalmente, este conocimiento también podría ser utilizado, según lo expuesto previamente, para corregir o reparar los modelos de referencia descubiertos en base a las actividades evidenciadas del entrenamiento de los usuarios. De esta forma, los modelos de referencia descubiertos son enriquecidos con el conocimiento del instructor experto que diseña el escenario.

### 4.1.3. Verificación

La etapa final de la metodología implica la verificación del entrenamiento de los participantes al realizar comparaciones con modelos de referencia. Es decir, dado un modelo de referencia que representa la actividad ideal necesaria para cumplir con el entrenamiento, se buscan detectar desviaciones con la realidad registrada en el log de eventos que recopila efectivamente la actividad realizada por los participantes.

En particular, para esta tarea también se utiliza el *plugin* Inductive Miner de la herramienta ProM Tools. Dicho *plugin* permite identificar desviaciones [72] entre un modelo y un log, y viceversa, en base al cálculo de alineamientos. Si una traza contiene un evento que no está representado o permitido por el modelo, se trata de un movimiento del log de eventos; en cambio, si el modelo

requiere un evento que no fue registrado en el log de eventos, se trata de un movimiento del modelo [73].

La aplicación de esta verificación de conformidad tiene foco en la perspectiva de flujo de control, siendo posible identificar tareas que no son ejecutadas por los participantes o tareas que son ejecutadas fuera de orden. Esto podría sugerir que el participante no logró completar el entrenamiento o que lo hizo de una nueva forma que anteriormente no había sido registrada. En esta comparación también se podrían identificar tareas que son ejecutadas un mayor número de veces a lo normal, o por el contrario un menor número de veces.

El análisis comparativo con el modelo de referencia también se podría enfocar en la perspectiva de tiempo. Esto podría permitir que el instructor identifique participantes que resolvieron el escenario de forma muy rápida respecto a lo esperado, lo cual podría significar que el escenario resulta muy sencillo para sus conocimientos y habilidades, o que el participante hizo trampa al entrenarse. Alternativamente, se podrían identificar participantes que requieren de mucho tiempo para cumplir con el entrenamiento, lo cual podría denotar que el escenario resulta extremadamente complejo.

## 4.2. Aplicación de la metodología según tipos de escenarios

La metodología propuesta permite la evaluación del entrenamiento de los participantes en un *cyber range*. Sin embargo, debe ser instanciada según el tipo de escenario con el cual se está trabajando. Es decir, la metodología tendrá variaciones según se trabaje con escenarios ofensivos (*red team*) o escenarios defensivos (*blue team*). Estas variaciones se dan en la identificación de los eventos de interés a partir de la aplicación de las reglas del SIEM.

Como parte del diseño del escenario, el instructor debe definir analíticas o reglas del SIEM para identificar cierto tipo de actividad, dependiendo del escenario en cuestión. Adicionalmente, el instructor debe etiquetar las reglas en base a tácticas y técnicas de los *frameworks* MITRE ATT&CK o MITRE D3FEND. En escenarios ofensivos, en los cuales el participante toma el rol del atacante y debe efectuar ataques contra la infraestructura de entrenamiento, las reglas buscarán identificar dichos ataques y deben estar etiquetadas según el MITRE ATT&CK. Continuando con el caso ejemplificado en la Figura 4.2,

las actividades mostradas en el modelo coinciden con tácticas del *framework* MITRE ATT&CK. La táctica de acceso a credenciales justamente implica que el atacante obtiene credenciales que le permiten acceder a un sistema, y puede estar representada por ataques de fuerza bruta donde el participante realiza múltiples intentos de autenticación probando distintas combinaciones de nombre de usuario y contraseña ante cierto sistema, por ejemplo en una máquina Linux ante el servicio SSH. Por otra parte, la táctica de descubrimiento implica que el atacante obtiene información sobre un sistema para posteriormente desarrollar otros ataques, y se puede ver representada por acciones que realiza el participante sobre la máquina como son: listar procesos en ejecución y conexiones de red activas, identificar usuarios locales de la máquina, entre tantas otras tareas que le permitan reconocer y obtener información sobre la máquina.

En cambio, en escenarios defensivos, donde el participante toma el rol del defensor y debe efectuar tareas defensivas como la identificación de vulnerabilidades, mitigación de incidentes y configuración de controles de seguridad, entre otras; las reglas intentarán identificar estas actividades. En este caso las reglas deben estar etiquetadas en base al MITRE D3FEND.

En el contexto de escenarios ofensivos, donde la idea es identificar la actividad maliciosa llevada adelante por el participante, el uso de reglas de SIEM resulta natural. Esto se explica en que justamente, el objetivo primordial de un SIEM es identificar este tipo de actividad. Para el diseño de este tipo de reglas se pueden utilizar recursos como el Cyber Analytics Repository (CAR) [74]. Este proyecto desarrollado por el MITRE especifica analíticas de seguridad para identificar actividad maliciosa basadas en el MITRE ATT&CK [75]. Estas reglas se especifican en base a pseudocódigo de forma que pueden ser traducidas a distintos lenguajes utilizados por distintas herramientas de SIEM.

Sin embargo, en el contexto de escenarios defensivos, la definición e implementación de reglas para la identificación de actividad defensiva se puede tornar más compleja. Esto se debe a que la función de un SIEM no es la identificación de actividad defensiva. Sin embargo, como parte de nuestro trabajo proponemos utilizar el SIEM para la detección de actividad defensiva. Tanto la actividad ofensiva como defensiva implican acciones que modifican el estado de un sistema. Dado que en un SIEM se tiene completa visibilidad sobre distintos eventos que ocurren en un sistema, entendemos que es posible lograr identificar la actividad defensiva ejecutada por el participante en su entrenamiento.

En síntesis, nuestra propuesta implica extender el enfoque de la herramienta de SIEM para incluir además la identificación de actividad defensiva. En este caso, la generación de reglas por parte del instructor cobra un papel de vital importancia ya que no hemos identificado repositorios de analíticas defensivas para un SIEM. Por lo tanto, el instructor deberá implementar reglas de forma manual para cada escenario y etiquetarlas según las tácticas y técnicas del *framework* MITRE D3FEND.

De acuerdo con lo mencionado anteriormente, una vez que se tienen identificados los eventos de interés se construye el log de eventos. Cada entrada del log tendrá: un tiempo de ejecución, un caso asociado y la actividad. La intención es que esta actividad queda expresada en términos de las tácticas y técnicas de los *frameworks*. En definitiva, el log de eventos recopila la actividad ejecutada por los participantes en su entrenamiento de forma que es detallada según tácticas y técnicas del *framework* MITRE ATT&CK o MITRE D3FEND (según el tipo de escenario). En la Tabla 4.1 se tiene un ejemplo de un log de eventos que cumple estas características. Posteriormente, los modelos construidos a partir de este log permiten visualizar el entrenamiento según las tácticas y técnicas aplicadas. Es decir, que permiten mostrar los ataques ejecutados o las acciones defensivas aplicadas según tácticas y técnicas de estos *frameworks*.

El uso de estos *frameworks* se justifica mediante dos razones primordiales. En primer lugar, al emplear estos *frameworks* se está reutilizando todo el conocimiento generado por una comunidad de expertos que se encargó de caracterizar y especificar actividad ofensiva y defensiva. El uso de estas tácticas y técnicas permite expresar la actividad de los participantes en su entrenamiento de una forma estandarizada que facilita su análisis y comprensión. Adicionalmente, permite contextualizar la actividad realizada por el participante y comprenderla como parte del proceso de ataque o defensa. Generalmente, existe una relación de precedencia entre las tácticas de cada *frameworks*. A modo de ejemplo, previo a la exfiltración de información (implica el robo de información desde una red) es necesario recolectar la información (obtener la información desde fuentes como bases de datos). Por lo tanto, al identificar las tácticas y técnicas empleadas, de cierta forma se está identificando en qué instancia del proceso de ataque o defensa se encuentra el actor.

Como segunda razón, se tiene que es posible expresar la actividad de los participantes y generar modelos en base a distintos niveles de granularidad, según

se utilicen las tácticas o las técnicas de los *frameworks* MITRE ATT&CK o MITRE D3FEND. En una primera instancia, los modelos podrían ser generados únicamente en base a las tácticas, las cuales representan los objetivos a alto nivel de los atacantes o defensores. Es decir, estos modelos permitirían entender las actividades a alto nivel llevadas a cabo por los participantes en su entrenamiento. En caso de que el instructor requiera un análisis más detallado, puede optar por expresar los modelos en base a las técnicas, las cuales detallan cómo los atacantes o defensores cumplen con las tácticas. Es decir que este segundo análisis permitiría entender cómo realizó el entrenamiento el participante, identificando actividades en concreto llevadas a cabo para cumplir con los objetivos de entrenamiento. Sin embargo, es importante tener en cuenta que al expresar las actividades en base a técnicas, dado que son un número mucho mayor que las tácticas, es esperable que el log de eventos crezca en cantidad de eventos y por lo tanto la aplicación de las tareas de minería de proceso resulte más costosa.

## 4.3. Casos de uso

En esta sección se presentan casos de uso de la metodología de evaluación, diferenciados por el objetivo final por el cual son aplicados.

### 4.3.1. Análisis de entrenamiento de participantes

El caso de aplicación más claro de la metodología es la evaluación del entrenamiento de los participantes. Con la aplicación de esta metodología, el instructor puede construir modelos que representan la actividad llevada a cabo por los participantes, expresados de forma estandarizada a través del uso de tácticas y técnicas de los *frameworks* MITRE ATT&CK y MITRE D3FEND.

Teniendo estos modelos disponibles, el instructor puede realizar distintos tipos de análisis para comprender si el estudiante cumple con los objetivos de entrenamiento del escenario y cómo realiza su entrenamiento. El análisis se puede realizar en base a distintas perspectivas. Una de estas perspectivas es el flujo de control, es decir, en las actividades ejecutadas y su orden y frecuencia de ejecución. Otra de las perspectivas de análisis es el tiempo. En este caso, el análisis se centra en el tiempo consumido por los participantes para completar el entrenamiento o tiempos consumidos por cada actividad. De esta forma, el

instructor puede llegar a identificar participantes que completan el entrenamiento, participantes a los cuales cierta tarea les consumió mucho tiempo o requirieron de muchos intentos para ejecutarla, lo cual podría indicar un nivel de dificultad desmedido. De forma contraria, puede identificar participantes que avanzan en el escenario ejecutando tareas en muy pocos intentos o consumiendo poco tiempo, lo cual podría indicar que dichas actividades resultan muy sencillas para los participantes. En el ejemplo presentado con la Figura 4.2 si se trata de un escenario ofensivo donde el objetivo final es obtener privilegios de *root* sobre una máquina Linux y que involucra: un ataque de fuerza bruta para la obtención de credenciales de acceso, posterior aplicación de tareas de descubrimiento para la obtención de información de la máquina y por último ejecución de un ataque de escalada de privilegios para lograr acceso de *root*, entonces analizando el modelo el instructor podría determinar que al no verse representada la última etapa de escalada de privilegios el participante no completó el escenario de forma satisfactoria, aunque si logró un avance hasta la etapa de descubrimiento.

Adicionalmente, la actividad de los participantes podría ser comparada en base a modelos de referencia que representan la actividad ideal del entrenamiento. A través de este análisis, el instructor podría llegar a identificar casos en los que los participantes cumplen de forma satisfactoria con el entrenamiento y comprender las actividades y técnicas aplicadas por estos. De forma alternativa, el instructor podría identificar casos en los que los participantes no cumplen los objetivos de entrenamiento y, por ejemplo, comprender si esto ocurre porque el escenario resulta muy complejo para las habilidades de los participantes, ya sea porque una actividad se repite muchas veces o porque dura mucho en su ejecución.

En resumen, el análisis realizado por la aplicación de esta metodología podría resultar de utilidad para que el instructor pueda brindar *feedback* al participante sobre su entrenamiento en el *cyber range*.

### 4.3.2. Detección de trampa

La aplicación de la metodología también podría estar enfocada en la identificación de trampas por parte de los participantes. Un participante comete trampa cuando entrega la *flag* final que fue obtenida porque otro participante le indicó cuál era el valor o porque le indicó cuáles son los pasos necesarios

para obtenerla. En otras palabras, un participante comete trampa cuando su entrenamiento no es individual y está guiado por la experiencia de otro participante. Generalmente, el primer tipo de trampa puede ser mitigado utilizando valores distintos para las *flags* de cada participante, como es posible de realiza en Tectonic ya que permite individualizar las *flags* de cada instancia del escenario.

La idea sería comparar el proceso generado por el participante en su entrenamiento con un modelo de referencia que represente las actividades mínimas y necesarias para cumplir con el entrenamiento. En caso de que en estas comparaciones se identifiquen desviaciones, se podría llegar a entender que el participante ha realizado trampa, dado que, por ejemplo, cumple con los objetivos de entrenamiento (entrega las *flags* del escenario), pero no realiza actividades clave que aparecen en el modelo de referencia o las ejecuta en un orden distinto. Para realizar esta tarea resulta crítica la construcción del modelo de referencia, pues este debe ser un modelo completo en el cual se tenga un alto nivel de confianza para lograr identificar trampas por parte de los participantes evitando generar falsos positivos.

Por otra parte, se podría llegar a pensar que si un participante ejecuta las tareas exactas para la resolución del escenario sin repetirlas o demora muy poco tiempo en resolver el escenario es porque ya conocía la solución, por ejemplo, porque otro participante le indicó cómo resolver el entrenamiento. Por lo tanto, el análisis de la cantidad de actividades ejecutadas y el tiempo total de ejecución del proceso de un participante puede ser de interés para el instructor, con el fin de compararlos con los procesos de los restantes participantes.

### 4.3.3. Análisis del diseño del escenario

La metodología también podría utilizarse para analizar el diseño del escenario e identificar mejoras en base al rendimiento de los participantes.

Al analizarse el entrenamiento de los participantes, se podría llegar a identificar actividades que se repiten muchas veces en forma de *loop* lo cual podría significar que el participante requiere de varios intentos para lograr avanzar en el escenario. También se podría identificar participantes que demoran mucho tiempo para completar el escenario. Esto podría ser un indicio que el escenario resulta de una complejidad excesiva para las habilidades de los participantes o que los conceptos teóricos asociados al escenario no fueron comprendidos,

siendo necesario aplicar ajustes directamente en el escenario o incluso en el componente teórico de un curso.

Por otra parte, el instructor podría identificar casos en los que participantes completan los objetivos de entrenamiento pero no siguen el modelo de referencia previamente definido. Esto podría significar que los participantes identifican nuevas formas de resolución del escenario que no fueron consideradas por el instructor. Esto se podría dar porque posterior al diseño del escenario se encontraron nuevas vulnerabilidades de sistemas y tecnologías utilizados en el escenario que no se conocían al momento que el instructor lo ideó. El participante podría aprovechar estas vulnerabilidades y explotarlas a su favor para cumplir con los objetivos del escenario "rompiendo" el flujo de resolución esperado por el instructor.

A modo de ejemplo, se podría pensar en un escenario donde se quiere tratar el concepto de *shell* reversa, una conexión iniciada desde la máquina víctima a la atacante que le brinda al actor malicioso una consola en la máquina atacada para la ejecución de comandos. Este escenario podría ser implementado a través de dos máquinas: la atacante y la víctima. En particular, esta última podría tener una aplicación web vulnerable que permite la subida de cualquier tipo de archivos sin ningún tipo de control. El usuario podría subir un archivo malicioso que al ser invocado establezca la *shell* reversa. Una vez establecida, el usuario puede ejecutar comandos en la máquina víctima y acceder a la *flag* que se encuentra en un archivo de la máquina víctima. Sin embargo, si dicha aplicación web tuviera vulnerabilidades de tipo *directory traversal* y Local File Inclusion (LFI), que no fueron identificadas por el instructor al momento de especificar el escenario, sería posible para el participante explotarlas para que dicho archivo de *flag* sea servido por el propio servidor web, es decir, que sea accedido directamente por el navegador web sin la necesidad del uso de la *shell* reversa y por lo tanto no siguiendo el flujo de resolución ideado por el instructor.

En conclusión, esta información puede resultarle útil al instructor con el objetivo de rediseñar el escenario para que se ajuste de mejor forma a los conceptos teóricos que quieren ser transmitidos y enseñados, y adicionalmente asegurando que los escenarios tengan un nivel de dificultad adecuado para el nivel de conocimiento y habilidades de los participantes que realizan el entrenamiento.

# Capítulo 5

# Experimentación

En este capítulo se presentan dos casos de aplicación de la metodología en entrenamientos ofensivos y defensivos realizados en Tectonic. Antes de presentar los casos de aplicación es importante aclarar como se instancia la metodología de evaluación propuesta al caso particular del *cyber range* Tectonic.

## 5.1. Instanciación de la metodología en Tectonic

Para instanciar la metodología en Tectonic (u otro *cyber range*) se debe definir esencialmente la tecnología de SIEM utilizada para la generación de eventos de interés y su categorización en base a las tácticas y técnicas de los *frameworks* del MITRE.

En el caso de Tectonic elegimos la herramienta de SIEM Elastic Security debido a varias razones. En primer lugar, es una herramienta *open source* y que cuenta con una licencia gratuita. Con esta licencia es posible cubrir las funcionalidades básicas necesarias para este trabajo que son: monitoreo en tiempo real de la infraestructura de entrenamiento y el motor de correlación y análisis de los eventos recolectados para la identificación y categorización de los eventos de interés.

Por otra parte, es una tecnología ampliamente utilizada en el ámbito de seguridad como solución de SIEM y que cuenta con una comunidad madura que brinda soporte al uso de la herramienta. Otra de las razones de la utilización de esta herramienta es nuestro conocimiento y experiencia previa sobre esta.

Además, la herramienta cuenta con un exhaustivo repositorio de reglas de

SIEM disponible para su uso. Como fue explicado, las reglas de SIEM son un punto fundamental para la identificación y categorización de los eventos de interés según los *frameworks* de MITRE. Este repositorio es mantenido por expertos en ciberseguridad que diseñan y mejoran reglas de forma constante. Al día de hoy, existen más de 1000 reglas de SIEM predefinidas, que permiten la detección de múltiples tipos de ataques y la gran mayoría de estas reglas ya se encuentran categorizadas según las tácticas y técnicas del *framework* MITRE ATT&CK. Al utilizar las reglas predefinidas por Elastic Security se está reutilizando el conocimiento generado por los expertos de seguridad que cuidadosamente diseñan e implementan estas reglas.

Como se mencionó con anterioridad, las máquinas donde se entrenan los estudiantes son monitoreadas utilizando el agente de Elastic, el cual permite recolectar distintos tipos de eventos de auditoría y seguridad, o mediante Packetbeat el cual permite la recolección de trazas de red. Todos los eventos recolectados son enviados al SIEM para su almacenamiento. Sin embargo, previo a su almacenamiento en Elastic Security, los eventos pasan por un proceso de normalización. Este proceso implica, por ejemplo, que los distintos campos y tipos de datos de los eventos son estandarizados y llevados a un formato común sin importar la tecnología o sistema que generó el evento. Por ejemplo, en un caso de eventos asociados a pedidos enviados a un servidor web que utiliza el protocolo HTTP, el método utilizado en el pedido se almacena en el campo de nombre *http.request.method* sin importar la tecnología utilizada como servidor web. Esto facilita el posterior análisis de los datos utilizando las reglas de SIEM.

Las reglas deben ser provistas por el docente como parte de la especificación del escenario. El docente puede definir sus propias reglas o puede decidir utilizar reglas predefinidas que vienen como parte de Elastic Security.

Por último, para la conformación del log de eventos, el número de instancia donde se evidenció el evento (información recopilada por el agente de Elastic o Packetbeat) es utilizado como identificador del caso. Dado que en Tectonic se despliegan instancias de escenarios, cada una asignada a un estudiante (o grupo de estudiantes) en particular; existe una relación uno a uno entre la instancia y el estudiante. Por lo tanto, el identificador de caso permite identificar al estudiante asociado a la actividad ejecutada y de esta forma llevar registro de toda la actividad ejecutada por cada estudiante como parte de su entrenamiento.

## 5.2. Casos de aplicación

Si bien en el contexto de este trabajo se monitorea la actividad y comportamiento de estudiantes y se recolectan estos datos, por temas de privacidad se recolectan los datos mínimos necesarios. Adicionalmente, los datos se encuentran seudonimizados para prevenir que sean asociados a un estudiante en particular. Por último, los estudiantes fueron notificados que estaban siendo monitoreados y se estaban recolectando datos de entrenamiento con el fin de investigación. En este trabajo no se publican datos personales que permitan identificar a los estudiantes.

### 5.2.1. Escenario ofensivo

En noviembre del año 2023 el GSI brindó un entrenamiento de carácter ofensivo a múltiples universidades de la región en el contexto de la Red Ciberlac. En total participaron siete equipos de estudiantes de grado en carreras asociadas a ingeniería en computación. La sesión de entrenamiento se desarrolló en un plazo de 3 horas, utilizando Tectonic como herramienta de despliegue (en particular el despliegue se realizó en la nube de AWS).

El ejercicio utilizado en este entrenamiento está conformado por dos máquinas, una máquina atacante y otra máquina víctima conectadas a través de una red. La máquina víctima ofrece los servicios File Transfer Protocol (FTP) y SSH. Al comienzo del ejercicio, los estudiantes cuentan con acceso total en la máquina atacante y su objetivo final es obtener privilegios de *root* (máximo nivel de privilegios en sistemas Linux) en la máquina víctima.

Para cumplir con dicho objetivo, los estudiantes deben ejecutar un ataque compuesto de los siguientes pasos claves:

- Realizar un ataque de fuerza bruta *online* al servicio FTP utilizando la herramienta Hydra [76]. Este ataque implica probar múltiples combinaciones de credenciales (usuario y contraseña) directamente realizando intentos de autenticación contra el servicio, hasta identificar credenciales correctas. El objetivo es obtener un usuario con acceso a un archivo de respaldo que contiene *hashes* de contraseñas de otros usuarios del sistema.
- Realizar un ataque de *password cracking* utilizando la herramienta John The Ripper [77] y el archivo obtenido en el paso anterior. Esta herramien-

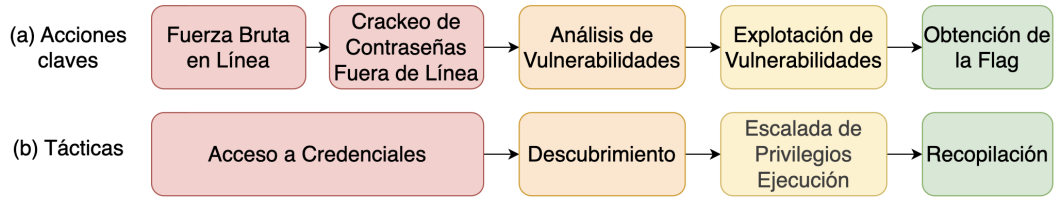



**Figura 5.1:** Entrenamiento ofensivo - Modelo de referencia

ta permite, en base a diccionarios o reglas de transformación, aplicar una función de *hash* a *strings* de entrada y comparar con un *hash* que ya se tiene. En caso de encontrar coincidencias en los valores de *hash* se encuentra la contraseña. Esto permite al estudiante obtener contraseñas de diferentes usuarios para el acceso SSH a la máquina víctima.

- Analizar vulnerabilidades y configuraciones inseguras en la máquina víctima. Esta actividad se puede realizar utilizando la herramienta LinPEAS [78].
- Explotar vulnerabilidades identificadas en el paso anterior para escalar privilegios y obtener acceso como *root* en la máquina víctima.
- Obtener la bandera que se encuentra en el archivo `/root/flag.txt`, sobre la cual solamente el usuario *root* tiene permisos de lectura.

La Figura 5.1 representa un modelo de referencia básico donde se exponen las actividades claves que deben realizar los estudiantes para cumplir con el entrenamiento, el orden esperado de ejecución y el mapeo de dichas actividades a tácticas del *framework* MITRE ATT&CK.

Para monitorear de manera efectiva las actividades de los estudiantes, se utilizó la opción de monitoreo de Tectonic implementada a través de agentes de Elastic, lo cual brinda completa visibilidad sobre las acciones ejecutados por los estudiantes en las máquinas de entrenamiento. Esta herramienta captura varios eventos de seguridad, como la ejecución de comandos y procesos, las conexiones de red, las autenticaciones de usuarios y la creación de archivos. A medida que estos eventos son capturados, se envían al SIEM para su análisis. El análisis y correlación de los eventos se realizó a través de la aplicación de diecisiete reglas de SIEM que buscaban identificar los eventos claves presentados anteriormente. De las diecisite reglas utilizadas, once se encontraban predefinidas por la herramienta Elastic Security y se enfocaban en la detección de ataques de fuerza bruta, enumeración de usuarios y otra información de

| Actividad | Porcentaje de ocurrencia |
|---|---|
| Acceso a credenciales | 81,454 % |
| Descubrimiento | 18,236 % |
| Escalada de privilegios | 0,234 % |
| Ejecución | 0,053 % |
| Recopilación | 0,023 % |

**Tabla 5.1:** Entrenamiento ofensivo - Distribución de eventos según la actividad

un sistema y ejecución de comandos maliciosos. Sin embargo, generamos seis nuevas reglas para la detección de actividad clave del estudiante que no era cubierta por las otras reglas. Estas reglas buscaban identificar: inicio de sesión exitoso desde la máquina atacante, la obtención de una *shell* con privilegios de *root* en la máquina víctima, obtención de claves Pretty Good Privacy (PGP) y descifrado de archivos utilizando dichas claves (estos pasos formaban parte de una posible escalada de privilegios) y la obtención de la *flag* final.

Todas las reglas utilizadas fueron catalogadas según las tácticas del *framework* MITRE ATT&CK. En total se utilizaron las siguientes cinco tácticas:

- Acceso a credenciales: el adversario intenta obtener credenciales.
- Descubrimiento: el adversario intenta obtener información sobre sistemas y la red interna para utilizar en otros ataques.
- Escalada de privilegios: el adversario intenta obtener mayores privilegios en un sistema.
- Ejecución: el adversario intenta ejecutar código malicioso.
- Recopilación: el adversario intenta obtener información de interés.

El log de eventos fue formado en base a los eventos de interés identificados con la aplicación de las reglas de SIEM. La fecha en la que ocurrió el evento se utilizó como *timestamp*, la táctica asociada como la actividad y el número de instancia que permite además identificar al grupo de estudiantes como el identificador de caso. En definitiva, este log recopila la actividad clave identificada para cada grupo en su entrenamiento en términos de las tácticas del *framework* MITRE ATT&CK. El log de eventos final consistió de 30.357 eventos divididos en cinco actividades o tácticas como se observa en la Tabla 5.1.

Las acciones más comunes se centraron en la obtención de credenciales de usuario, en concreto mediante ataques de fuerza bruta al servicio FTP o SSH

| Grupo | Cantidad de eventos |
|---|---|
| 1 | 3.965 |
| 2 | 775 |
| 3 | 35 |
| 4 | 12.505 |
| 5 | 1.165 |
| 6 | 11.858 |
| 7 | 54 |

**Tabla 5.2:** Entrenamiento ofensivo - Cantidad de eventos por grupo

del sistema de la víctima. Esto tiene sentido ya que el ataque implica el uso de herramientas automáticas que hacen múltiples intentos de autenticación en un servicio para la obtención de credenciales válidas. La mayoría de los grupos consideraron que esta parte del escenario fue la más complicada y la que más tiempo requirió. Por el contrario, la actividad menos ejecutada es la recolección asociada a la obtención de la bandera final. Esto se puede explicar por dos razones: en primer lugar, no todos los grupos lograron obtener la bandera, y en segundo lugar, para aquellos grupos que sí lo hicieron, la tarea no resulta complicada y se suele realizar en un único intento, dado que una vez se tiene privilegio de *root* simplemente se imprime el contenido del archivo.

En la Tabla 5.2 y Figura 5.2 se detallan la cantidad de eventos y duración total de cada grupo, es decir, los casos del proceso. Se logran identificar valores bastante diferentes para los distintos grupos. Por ejemplo, los grupos 3 y 7 tuvieron menos eventos porque las reglas del SIEM no detectaron gran parte de sus ataques de fuerza bruta. Esta falla se produjo porque las reglas fueron diseñadas para identificar dichos ataques solo cuando la cantidad de intentos de inicio de sesión fallidos dentro de un período de tiempo específico supera un umbral predeterminado. Dado que algunos grupos ejecutaron varios intentos de inicio de sesión que no superaron este umbral, sus ataques de fuerza bruta no se detectaron. Como resultado, la hora de inicio identificada para estos grupos no refleja la hora de inicio real del entrenamiento y en consecuencia la duración total del proceso es mucho menor en comparación con los otros grupos.

En cuanto al grupo 1, que presentó la menor duración del proceso, del análisis realizado se obtuvo que, si bien hay evidencia de un ataque de fuerza

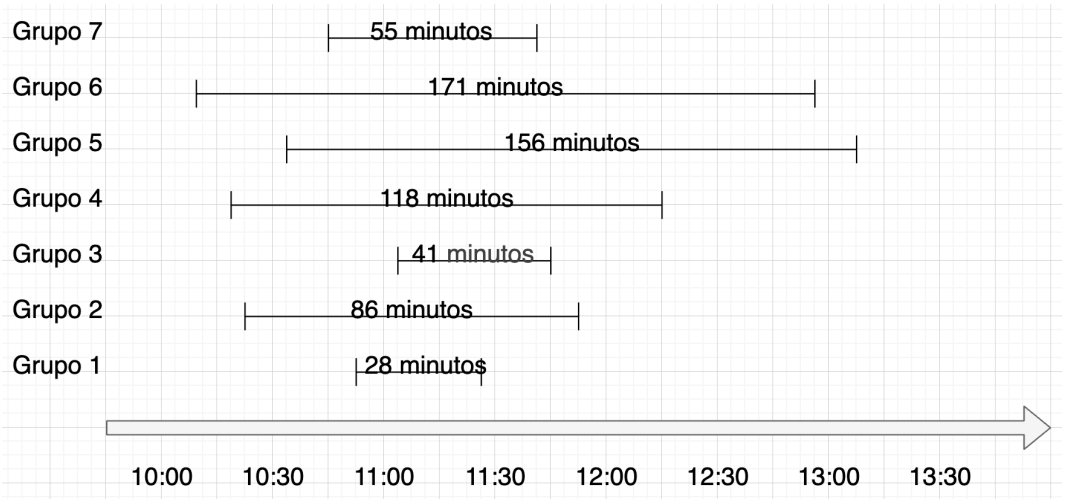



**Figura 5.2:** Escenario Ofensivo - Duración por grupo

bruta y de un acceso exitoso a la máquina víctima, hay pocos registros de actividades de descubrimiento en la máquina víctima y ninguna evidencia de tareas relacionadas con la escalada de privilegios. Esto sugiere que esta segunda fase del ataque o proceso pudo resultar de una mayor complejidad para el grupo, lo que llevó a un progreso mínimo. Dado que no hay registros de estas actividades, su proceso "termina antes" y por lo tanto resultó en la duración más corta en comparación con otros grupos.

Finalmente, los grupos 4, 5 y 6, los grupos de mayor duración, mostraron una variedad de comportamientos. El grupo 4 completó la práctica satisfactoriamente, mientras que los grupos 5 y 6 no lo hicieron. En concreto, el grupo 6 no pudo obtener un usuario del sistema, por lo que su ataque de fuerza bruta no tuvo éxito. Su proceso consistió únicamente en la actividad asociada con el intento de obtener credenciales.

Se aplicó descubrimiento de procesos para crear un modelo que representa el entrenamiento de los estudiantes basado en las tácticas del MITRE ATT&CK. El modelo resultante se presenta en la Figura 5.3, que muestra el punto de inicio (punto verde), el punto final (punto rojo) y todas las actividades que realizan los estudiantes. Los números asociados a las actividades y rutas indican la cantidad de ejecuciones, y el color azul más oscuro representa las actividades y rutas que se ejecutan una mayor cantidad de veces. Para una mayor descripción de la notación utilizada por los modelos presentados en este trabajo referirse al Anexo 1.

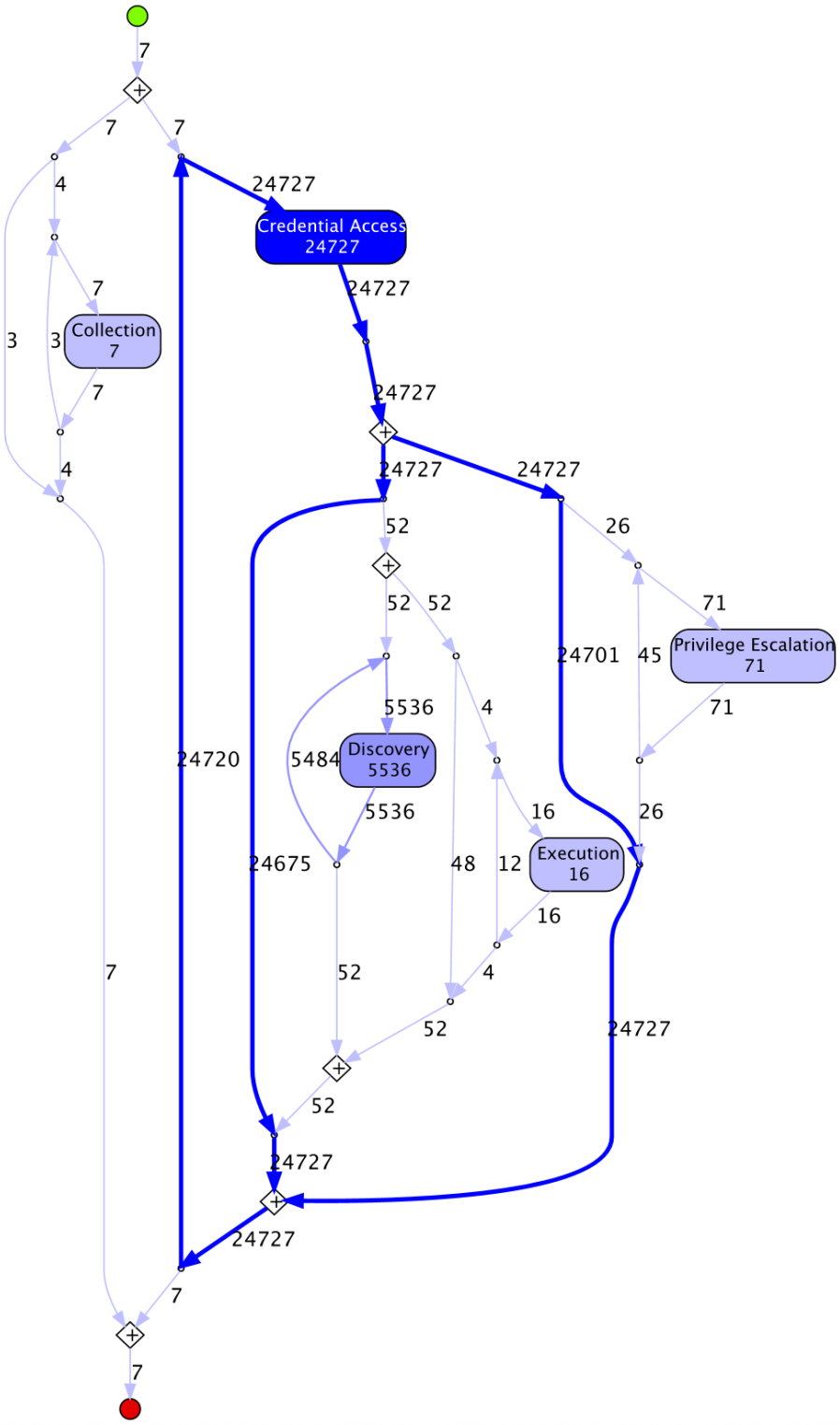
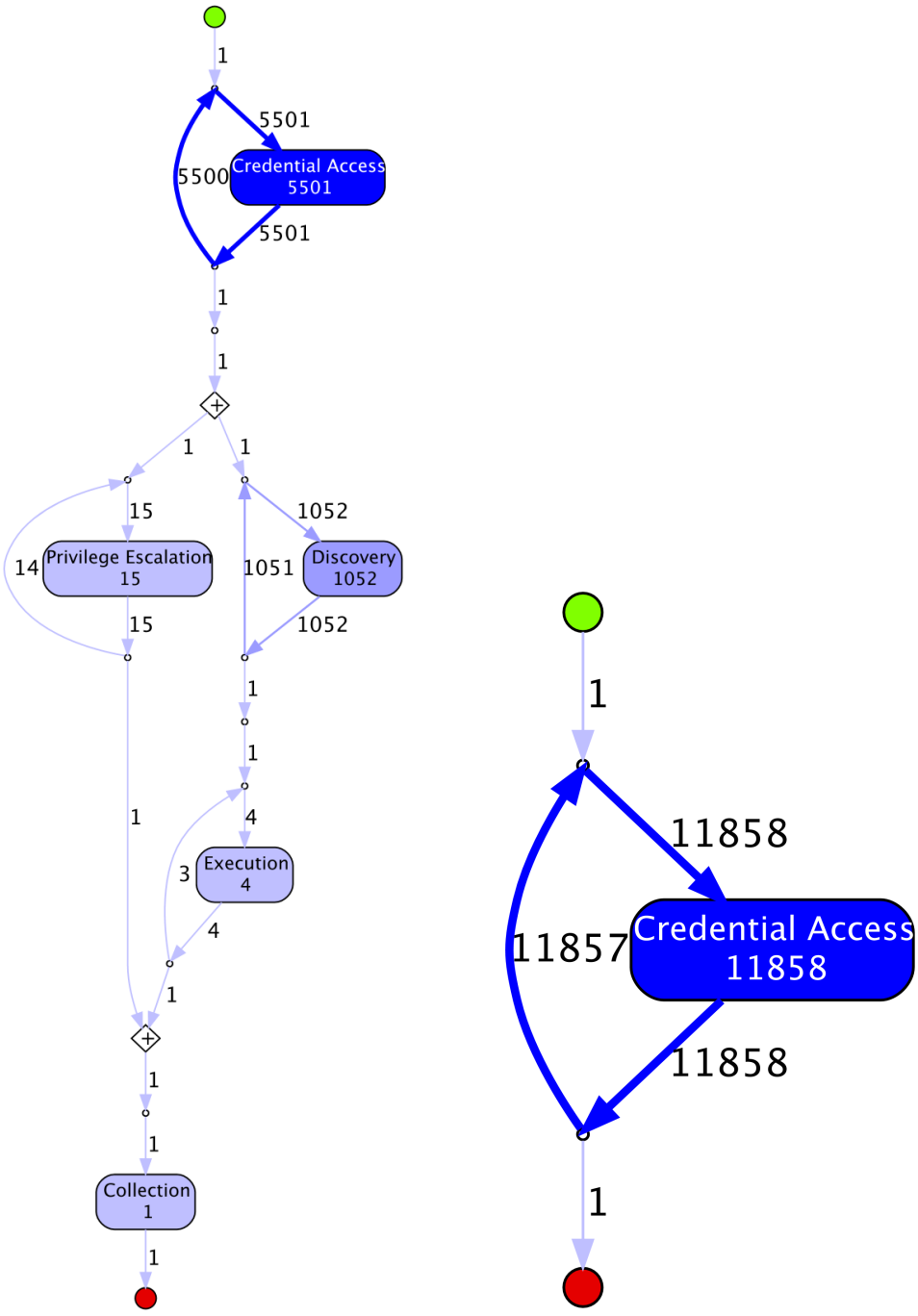



**Figura 5.3:** Entrenamiento ofensivo - Modelo del proceso de entrenamiento

Según el modelo, la actividad más ejecutada en el proceso es el acceso a credenciales, que se realiza típicamente en bucle. Esto se debe a la naturaleza de los ataques de fuerza bruta, que requieren múltiples intentos con diferentes contraseñas hasta que se identifica una válida. La segunda actividad más ejecutada es el descubrimiento, que implica obtener información sobre la máquina víctima para ser utilizada posteriormente para el ataque de escalada de privilegios. Para llevar a cabo esta actividad, los grupos utilizaron la herramienta automática LinPEAS, que realiza numerosas tareas de descubrimiento asociadas en una única ejecución. La actividad de recolección, que implica la obtención de la bandera final, fue la acción menos ejecutada. Solo cuatro grupos ejecutaron esta acción, y solo tres completaron la práctica, lo que indica un falso positivo en uno de los casos alertados y la necesidad de mejoras en la implementación de la regla. El análisis del modelo revela que las tareas de descubrimiento, escalada de privilegios y ejecución ocurren después de al menos un evento de acceso a credenciales. Esto se alinea con el comportamiento esperado porque, en la primera etapa del escenario, el estudiante debe ejecutar el ataque de fuerza bruta para obtener acceso a la máquina víctima. Luego, el estudiante realiza las tareas de descubrimiento, ejecución y escalada de privilegios para obtener el usuario *root*, y posteriormente el acceso a la *flag*.

Adicionalmente, se generaron modelos del proceso de entrenamiento de cada grupo, lo que permite un análisis más exhaustivo de sus actividades. Es importante aclarar que para la construcción de estos modelos se realizó un filtrado del log de eventos eliminando actividades que generan modelos de mayor complejidad y que resultan más difícil de analizar. Por ejemplo, en lo que tiene que ver con la táctica de acceso a credenciales, existen dos actividades básicas asociadas: la ejecución de ataques de fuerza bruta y el uso de las credenciales obtenidas a través de estos ataques. Dado que existen varios usuarios cuyas credenciales pueden ser comprometidas, es muy común que los estudiantes encuentren un primer usuario con el que acceden a la máquina víctima para aplicar tareas de descubrimiento y escalada de privilegios, pero a la vez continúan con ataques de fuerza bruta para identificar otros usuarios. Esto genera que las tácticas de acceso a credenciales se encuentren "mezcladas" con las otras tácticas como puede ser el descubrimiento. En definitiva, esto genera modelos más complejos. Por esta razón es que se aplicó un filtrado del log de eventos generado para la construcción de modelos para cada grupo de estudiantes. La Figura 5.4 ilustra el caso de los grupos 4 y 6.




**(a)** Grupo 4 **(b)** Grupo 6

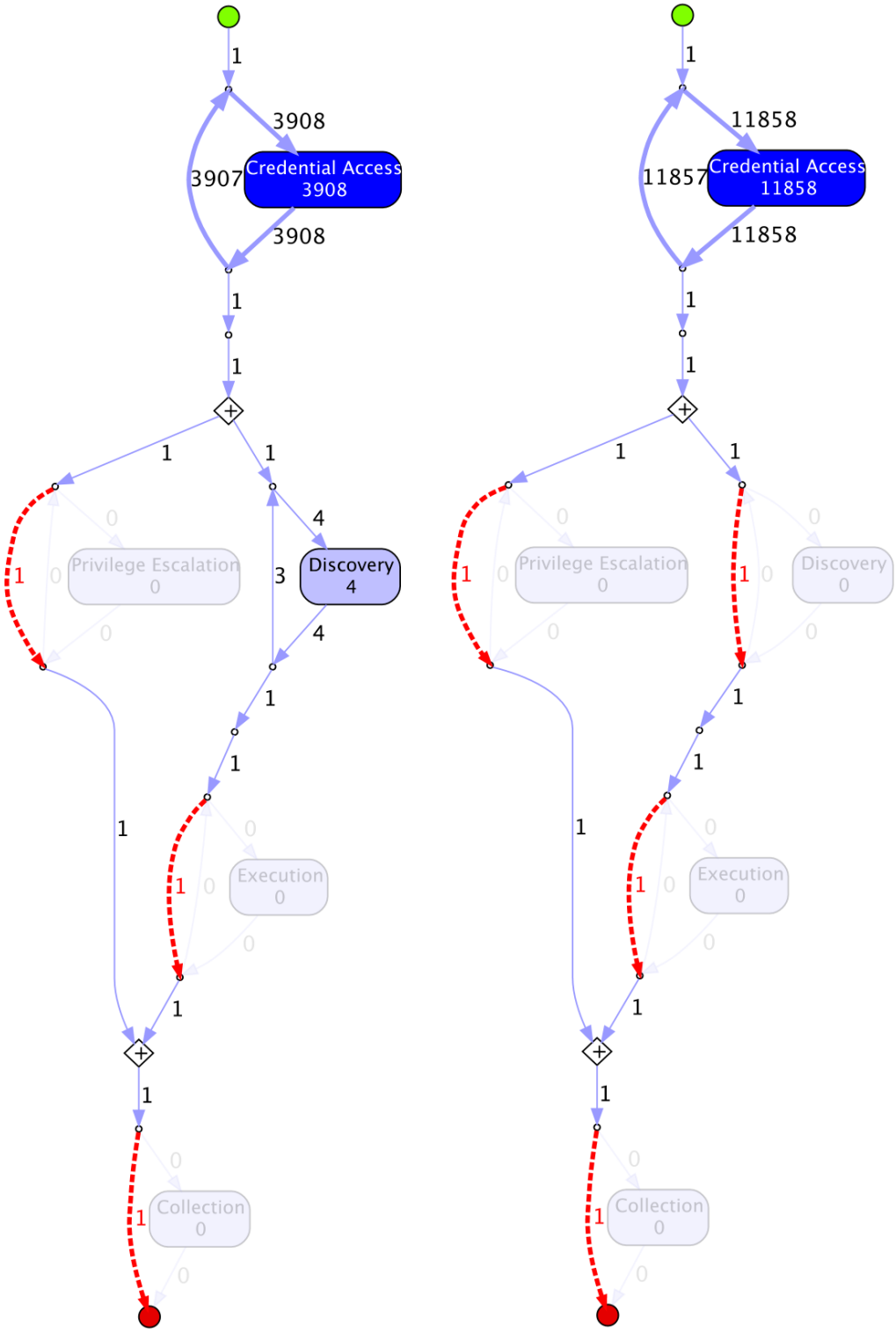
**Figura 5.4:** Entrenamiento ofensivo - Modelos de grupos 4 y 6

Estos modelos muestran comportamientos completamente distintos entre sí. El grupo 6 tuvo dificultades y no logró avanzar más allá del ataque de fuerza bruta, con solo registros de actividad relacionada con el acceso a credenciales. Por el contrario, para el grupo 4 sí se lograron identificar las distintas actividades que permiten la realización exitosa del entrenamiento. Al examinar los modelos, se identifica que los grupos 1 y 5 llevaron a cabo ataques de fuerza bruta y pudieron acceder a una cuenta de usuario en la máquina víctima. Sin embargo, a pesar de realizar tareas de descubrimiento en la máquina víctima, no pudieron obtener el usuario *root*. Por el contrario, los grupos 2 y 7 compartieron muchas similitudes, ya que resolvieron la práctica con éxito. Al igual que el grupo 4, se tiene registro de las distintas actividades ejecutadas que llevaron a la resolución correcta de la práctica. Por último, el caso del grupo 3 es particular porque si bien el modelo generado es muy similar a los modelos de los grupos 2, 4 y 7 que completaron satisfactoriamente el escenario, realmente el grupo 3 no lo hizo (esto se corroboró con el grupo en base a una entrevista). A raíz de esto se identificó un error en la regla del SIEM que alerta ante la obtención de la *flag*, pues el caso alertado del grupo 3 es un falso positivo.

A pesar de no contar con un modelo de referencia que detalle la actividad esperada para el cumplimiento del escenario, se decidió aplicar verificación de conformidad. En este sentido se realizó una comparación entre la propia actividad de los distintos grupos. Para esto se tomó como modelo de referencia el generado para el grupo 4, quien completó de forma satisfactoria el escenario, y se aplicó verificación de conformidad con el log de eventos formado exclusivamente con la actividad del grupo 1 y otro log de eventos con la actividad del grupo 6. Como ya fue detallado, tanto el grupo 1 como grupo 6 no completaron el escenario.

En la Figura 5.5 se pueden ver los resultados obtenidos. Las líneas rojas en la figura denotan desviaciones respecto al modelo y se observa que las actividades de escalada de privilegios, ejecución y recopilación no se encuentran marcadas ya que no fueron ejecutadas por los grupos 1 y 6.

Adicionalmente, se realizó una comparación "manual" con el modelo de referencia que se muestra en la Figura 5.1. Este modelo de referencia se creó manualmente, describiendo las tácticas empleadas y su respectiva secuencia de ejecución. Si bien este modelo de referencia puede no representar con precisión los intentos repetidos de los estudiantes para ejecutar las actividades, la comparación de dicho modelo y los modelos de los estudiantes que com-




**(a)** Grupo 1 vs Grupo 4

**(b)** Grupo 6 vs Grupo 4

**Figura 5.5:** Entrenamiento ofensivo - Verificación de conformidad

pletaron el entrenamiento muestra consistencia en la secuencia de actividades, en una primera instancia ejecutando ataques de fuerza bruta para el acceso a credenciales y una vez que se tenía acceso a la máquina víctima se sucedían actividades asociadas al descubrimiento, escalada de privilegios y obtención de la *flag* final.

En definitiva, verificación de conformidad resulta útil para identificar potenciales desviaciones entre la actividad efectivamente llevada a cabo por los usuarios en su entrenamiento respecto a un modelo de referencia que detalle la actividad esperada.

Es vital aclarar que todos los hallazgos derivados del análisis están respaldados por una sesión informativa realizada con los participantes del entrenamiento. En esta sesión, se discutieron varios aspectos, como el cumplimiento de los objetivos del entrenamiento por parte de los grupos, las actividades realizadas durante el entrenamiento y aquellas tareas que según el criterio de los participantes resultaron más complejas ya sea por cantidad de intentos o tiempo total de ejecución.

### 5.2.2. Escenario defensivo

Esta capacitación se llevó a cabo durante la III Escuela Latinoamericana de Informática (ELI), sucedida en agosto de 2024 en conjunto con la edición número 50 de la CLEI. La escuela contó con un curso titulado "Forensia Digital y Respuesta a Incidentes: un enfoque práctico" brindado por el GSI. Un componente importante del curso fue un laboratorio práctico en el que los estudiantes participaron en una simulación de un incidente de ciberseguridad.

El laboratorio presentó un escenario en el que una empresa de energía ficticia sufre un ciberataque dirigido a su infraestructura crítica de tipo Supervisory Control and Data Acquisition (SCADA)[1]. El escenario se basa en un laboratorio de ciberseguridad de European Union Agency for Cybersecurity (ENISA) [79]. De forma resumida, la idea del escenario propuesto es que la organización cuenta con una infraestructura constituida por múltiples redes y sistemas. En primer lugar, se cuenta con una red industrial, red que da soporte para la automatización de los distintos procesos y operativa de la empresa, con un terminal maestro para la gestión remota automatizada de múltiples sub-

[1]Los sistemas SCADA permiten controlar y supervisar procesos industriales de forma remota.

estaciones de generación de energía. Además, la organización mantiene una red de servicios que proporciona el servicio de correo, FTP y servicios web; junto con una red de monitoreo y una red corporativa donde se ubican los puestos de trabajo de empleados. El ataque involucra a un actor malicioso que obtiene acceso no autorizado a la terminal maestra. Los estudiantes deben responder a este incidente aplicando técnicas de detección, análisis, mitigación y recuperación, aprendidas en el componente teórico del curso.

El escenario se desplegó en AWS utilizando Tectonic. Si bien la infraestructura de la realidad planteada era compleja y extensa, solo se implementaron los componentes mínimos necesarios para el dictado del curso. En la Figura 5.6 se proporciona un diagrama de la infraestructura efectivamente desplegada. Cabe destacar que se crearon ocho copias de la infraestructura para acomodar a los ocho grupos de estudiantes. Los estudiantes accedieron a la infraestructura a través de una máquina de acceso para estudiantes designada, desde la cual podían conectarse a la terminal maestra a través del protocolo SSH. En el diagrama se observa una máquina atacante, desde la cual se compromete la terminal maestra. Inicialmente, los estudiantes desconocían la existencia de la máquina atacante y se esperaba que la identificaran como parte de su análisis. Posteriormente, debían implementar las estrategias de mitigación adecuadas para contener y recuperarse del ataque. Las instancias de entrenamiento fueron monitoreadas a través de la opción de monitoreo ofrecida por Tectonic que implica la instalación del agente de Elastic en las máquinas de entrenamiento.

Se utilizó el *framework* Caldera para simular, en tiempo real, un ataque sobre la infraestructura desplegada. El ataque fue diseñado por nosotros y consta de la siguiente secuencia de pasos:

- Ejecución de un ataque de fuerza bruta *online* contra el servicio SSH de la terminal maestra.
- Acceso a la terminal maestra a través de SSH utilizando las credenciales comprometidas en el paso anterior.
- Instalación de una puerta trasera dentro del servicio *cron* (servicio que gestiona la ejecución de tareas periódicas en sistemas Linux [80]), que establece una conexión de *shell* reversa desde la terminal maestra (máquina víctima) a la máquina atacante cada vez que se inicia el servicio.
- Ejecución de la *shell* reversa.
- Enumeración de la terminal maestra y recopilación de información con-

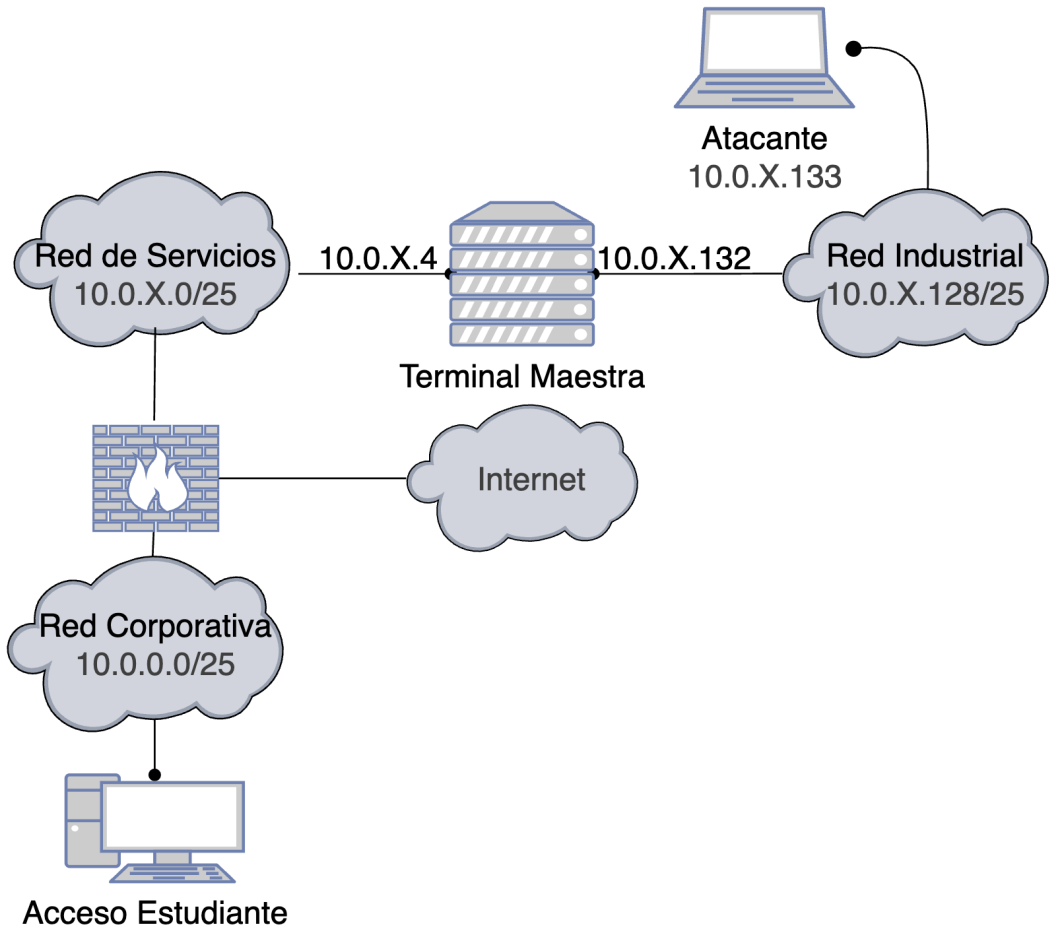



**Figura 5.6:** Escenario defensivo - Infraestructura del escenario

fidencial, la cual posteriormente es exfiltrada a través de la *shell* reversa.

Estas actividades generan trazas en los logs del sistema y otros artefactos en la terminal maestra. Parte de las tareas del estudiante es analizar dichas trazas y artefactos para comprender el ataque sucedido en la infraestructura de la organización.

El objetivo del estudiante en este escenario es responder eficazmente al incidente mediante la realización de un análisis forense y la aplicación de tareas de mitigación y recuperación. La Figura 5.7 ilustra las actividades clave que el estudiante debe realizar, junto con su catalogación según las tácticas del *framework* MITRE D3FEND.

Para detectar y analizar el ataque, se espera que los estudiantes examinen varios componentes, incluidos los procesos en ejecución, las conexiones de red establecidas, los logs del sistema, el servicio *cron* y la cuenta de administrador comprometida. Una vez que el estudiante comprende el ataque, debe imple-

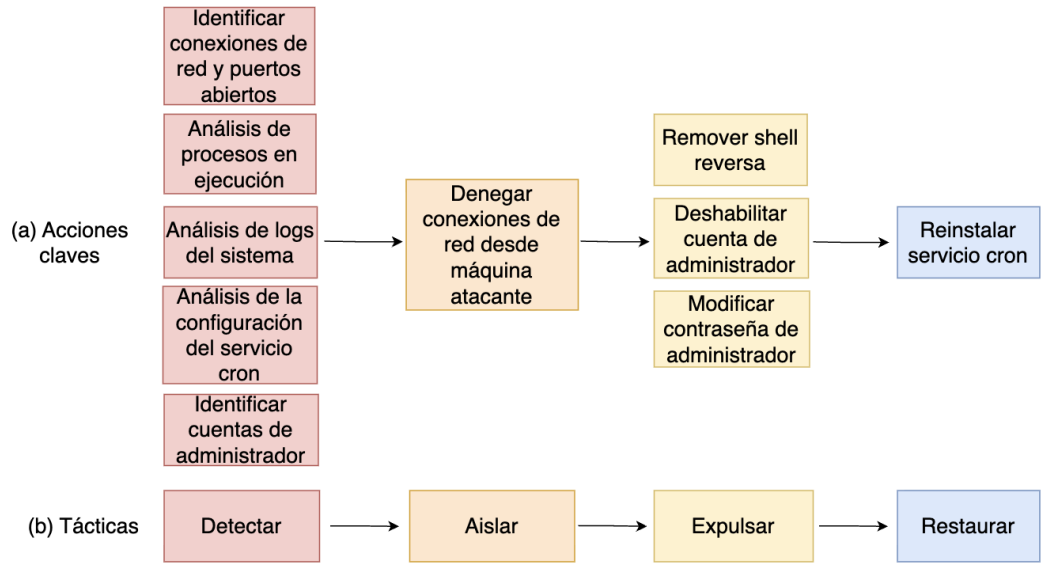



**Figura 5.7:** Escenario defensivo - Modelo de referencia

mentar medidas de mitigación y recuperación en la terminal maestra. Estas medidas pueden ser: bloqueo de conexiones de red entrantes y salientes a la máquina atacante, cambio de contraseña de la cuenta de administrador comprometida, eliminación de *shell* reversa y reinstalación del servicio *cron*. En definitiva, estas tareas pueden ser agrupadas en una de las siguientes cuatro tácticas del MITRE D3FEND:

- Detectar: identificar acceso no autorizado por parte de un adversario a un sistema o la red.
- Aislar: implica aislar los sistemas comprometidos por un adversario para evitar la propagación del ataque a otros sistemas.
- Expulsar: tareas asociadas a la remoción del adversario de un sistema o la red.
- Restaurar: recuperar un sistema a un estado de funcionamiento aceptable previo al ataque sufrido.

Dentro del SIEM se configuraron un total de dieciséis reglas diseñadas meticulosamente para correlacionar los eventos generados por los estudiantes al interactuar con la terminal maestra. Estas reglas identifican de manera efectiva las actividades clave descritas anteriormente. Las actividades identificadas por las reglas de SIEM se utilizaron para la construcción del log de eventos.

Se creó un log de eventos que recopila un total de 449 eventos. Inicialmen-

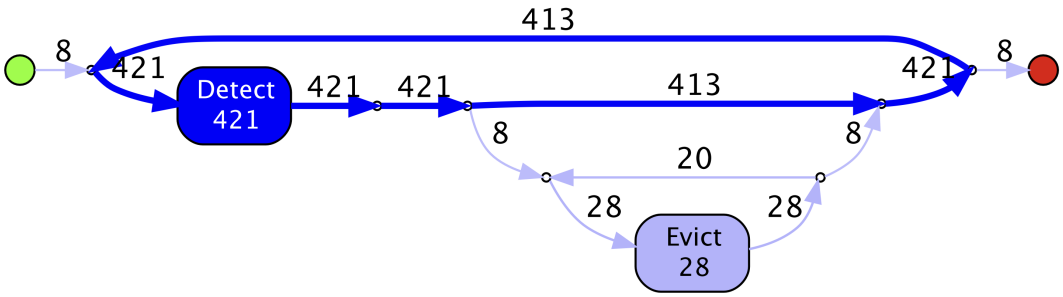



**Figura 5.8:** Escenario defensivo - Modelo del proceso de entrenamiento utilizando tácticas

| Actividad | Porcentaje de ocurrencia |
|---|---|
| Detectar - Análisis de sistema de archivo | 31,85 % |
| Detectar - Conexiones de red | 22,72 % |
| Detectar - Hash de archivos | 21,16 % |
| Detectar - Análisis de ejecución de procesos | 18,04 % |
| Expulsar - Deshabilitar cuentas de usuario | 3,12 % |
| Expulsar - Terminar procesos | 2 % |
| Expulsar - Remover archivos | 0,89 % |
| Expulsar - Revocar credenciales | 0,22 % |

**Tabla 5.3:** Escenario defensivo - Distribución de eventos según la actividad

te, el log utilizaba solamente las tácticas como actividades, lo que dio como resultado que se vieran representadas solamente dos actividades. En la Figura 5.8 se puede visualizar el modelo obtenido luego de aplicar descubrimiento de procesos a dicho log de eventos. El análisis a partir de este log de eventos brindaba un enfoque limitado y no proporcionaba el nivel de detalle necesario. En consecuencia, se optó por mejorar el log de eventos incorporando tanto las tácticas como las técnicas del *framework* MITRE D3FEND para representar las actividades de los estudiantes. Nuevamente, el tiempo de ejecución de cada actividad se utilizó como *timestamp* y el número de instancia que permite identificar al grupo que realiza la acción se utilizó como identificador del caso. Como resultado, se obtuvo un log de eventos más completo que presenta ocho actividades distribuidas de acuerdo con los datos presentados en la Tabla 5.3.

Se puede observar que los estudiantes solo ejecutaron tareas asociadas a las tácticas detectar y expulsar. En comparación con las actividades esperadas descritas en la Figura 5.7, las tácticas aislar y restaurar no fueron ejecutadas.

| Grupo | Cantidad de eventos |
|---|---|
| 1 | 20 |
| 2 | 49 |
| 3 | 104 |
| 4 | 69 |
| 5 | 65 |
| 6 | 67 |
| 7 | 36 |
| 8 | 39 |

**Tabla 5.4:** Entrenamiento defensivo - Cantidad de eventos por grupo

Además, las actividades de detección se ejecutaron con una mayor frecuencia, especialmente las asociadas con el análisis de registros del sistema, análisis de archivos de configuración y análisis de conexiones de red. Este énfasis se debe a la decisión de los instructores de priorizar estas actividades, ya que son más relevantes para realizar un análisis forense, un tema clave cubierto en la porción teórica del curso. En los momentos finales de la sesión de laboratorio, los instructores informaron a los estudiantes que podían aplicar actividades de mitigación y restauración, lo que puede explicar la ejecución limitada, y en algunos casos nula, de estas tareas.

En cuanto a las métricas de los grupos descritas en la Tabla 5.4 y Figura 5.9, se evidencia que los grupos ejecutaron entre 20 y 104 eventos, con un tiempo empleado que osciló entre 90 y 166 minutos. Algunos grupos, como el número 1 y el número 8, realizaron muy pocas tareas o requirieron un tiempo mínimo, concentrándose únicamente en actividades de detección. En cambio, la mayoría de los otros grupos pudieron implementar además tareas asociadas a la restauración de la infraestructura.

Descubrimiento de procesos fue empleado para la generación de un modelo que ilustra el proceso del entrenamiento de los estudiantes. Dicho modelo se encuentra representado en la Figura 5.10. El análisis de este modelo indica que las actividades que se realizan con más frecuencia son las relacionadas con la detección. Estas incluyen el análisis de procesos y conexiones de red y la revisión de archivos del sistema, como registros y archivos de configuración. Estas actividades relacionadas con la detección ocurren al principio del proceso de entrenamiento. En cambio, las actividades vinculadas a la táctica de expulsión se realizan con menos frecuencia; solo la mitad de los grupos implementaron esta tarea, y generalmente ocurren hacia el final del proceso. Este patrón es

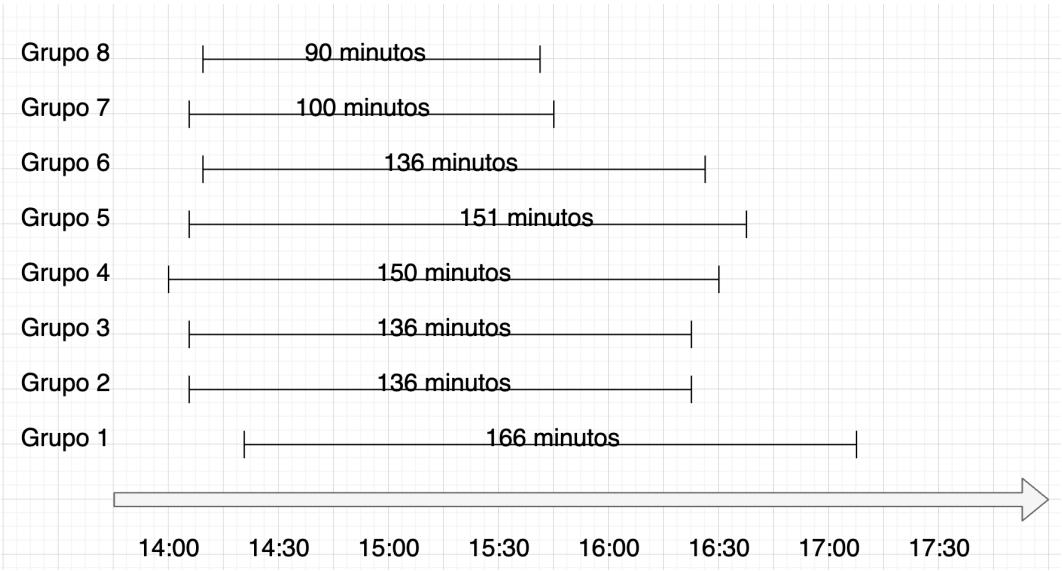



**Figura 5.9:** Escenario defensivo - Duración por grupo

comprensible, ya que estas acciones implican la finalización del proceso asociado con la *shell* reversa y el bloqueo o cambio de la contraseña de la cuenta de administrador comprometida. Los estudiantes primero debían comprender el ataque a través de actividades de detección para luego llevar a cabo estas tareas de expulsión del adversario de manera eficaz.

Adicionalmente, se generaron modelos particulares de cada grupo para analizar en mayor detalla la actividad de cada uno. En la Figura 5.11 se tiene como ejemplo el modelo generado para el grupo 6. Se puede observar que las actividades más ejecutadas por el grupo son aquellas asociadas a la táctica detectar en lo que tiene que ver con el análisis de logs, procesos ejecutando y conexiones de red activas. Las tareas asociadas a la táctica expulsar como son la eliminación de archivos, terminación de procesos y bloqueo de cuentas de usuarios se aplican en una menor medida y luego de tareas de detección.

Por último, una comparación manual con el modelo presentado en la Figura 5.7 revela que, aunque los estudiantes no emplearon dos tácticas, las tácticas que sí fueron aplicadas (específicamente detección y expulsión) se ejecutaron en el orden esperado. Es esencial aclarar la razón por la cual las actividades de eliminación de archivos y terminación de procesos aparecen "mezcladas" con las actividades de detección en el modelo. Estas actividades están vinculadas a la eliminación de la *shell* reversa. Si el estudiante simplemente detenía el proceso asociado con esta *shell*, el proceso se volvía a lanzar automáticamente.

Por lo tanto, se esperaba que el estudiante hiciera varios intentos para eliminar la *shell* reversa con éxito y, después de hacerlo, requiriera de más acciones de detección (como listar los procesos ejecutando) para confirmar que la eliminación fue efectiva. En definitiva, el orden de las acciones ejecutadas por los estudiantes se alinea al orden de ejecución esperado.

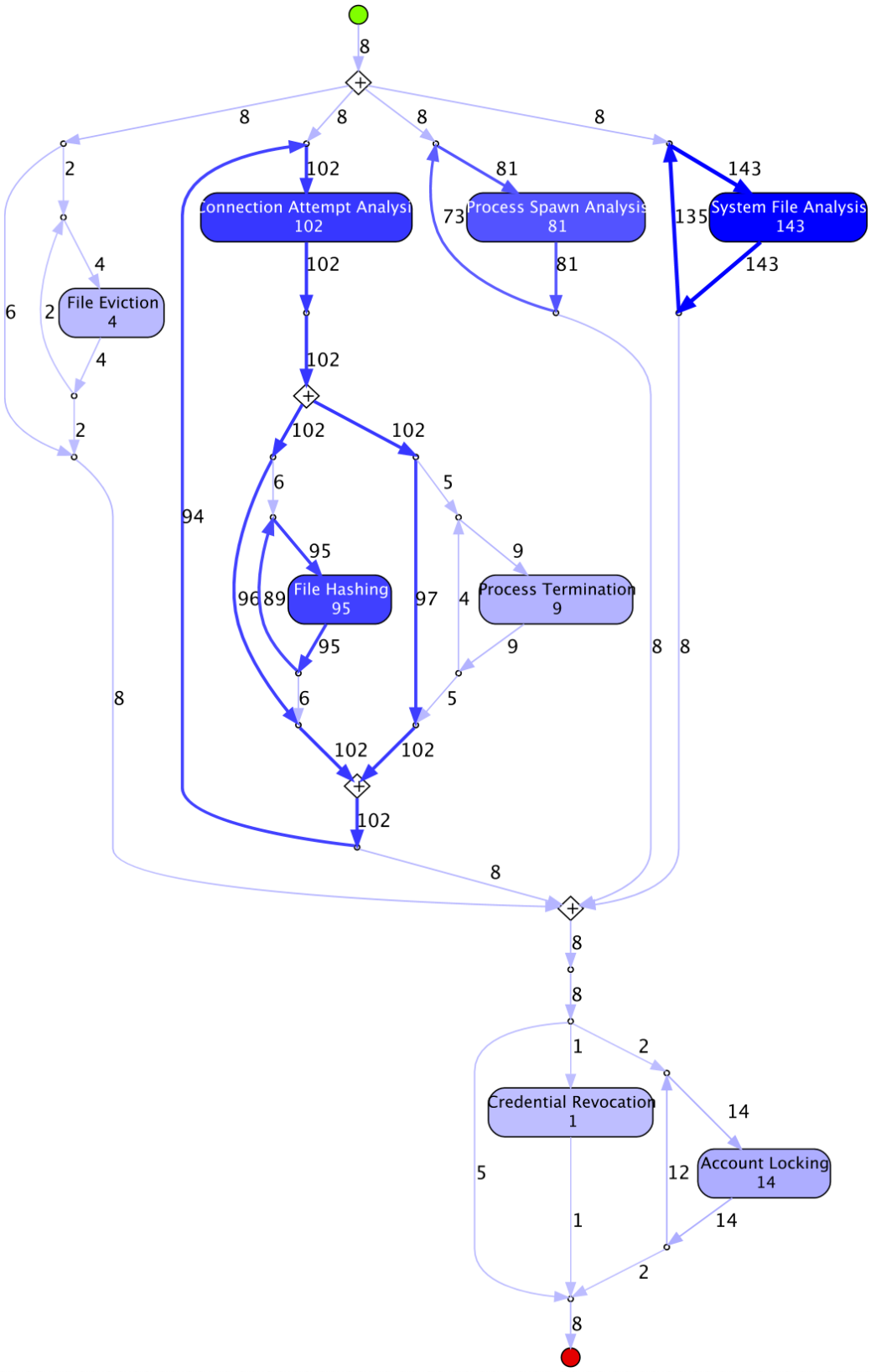



**Figura 5.10:** Escenario defensivo - Modelo del proceso de entrenamiento utilizando técnicas del MITRE D3FEND

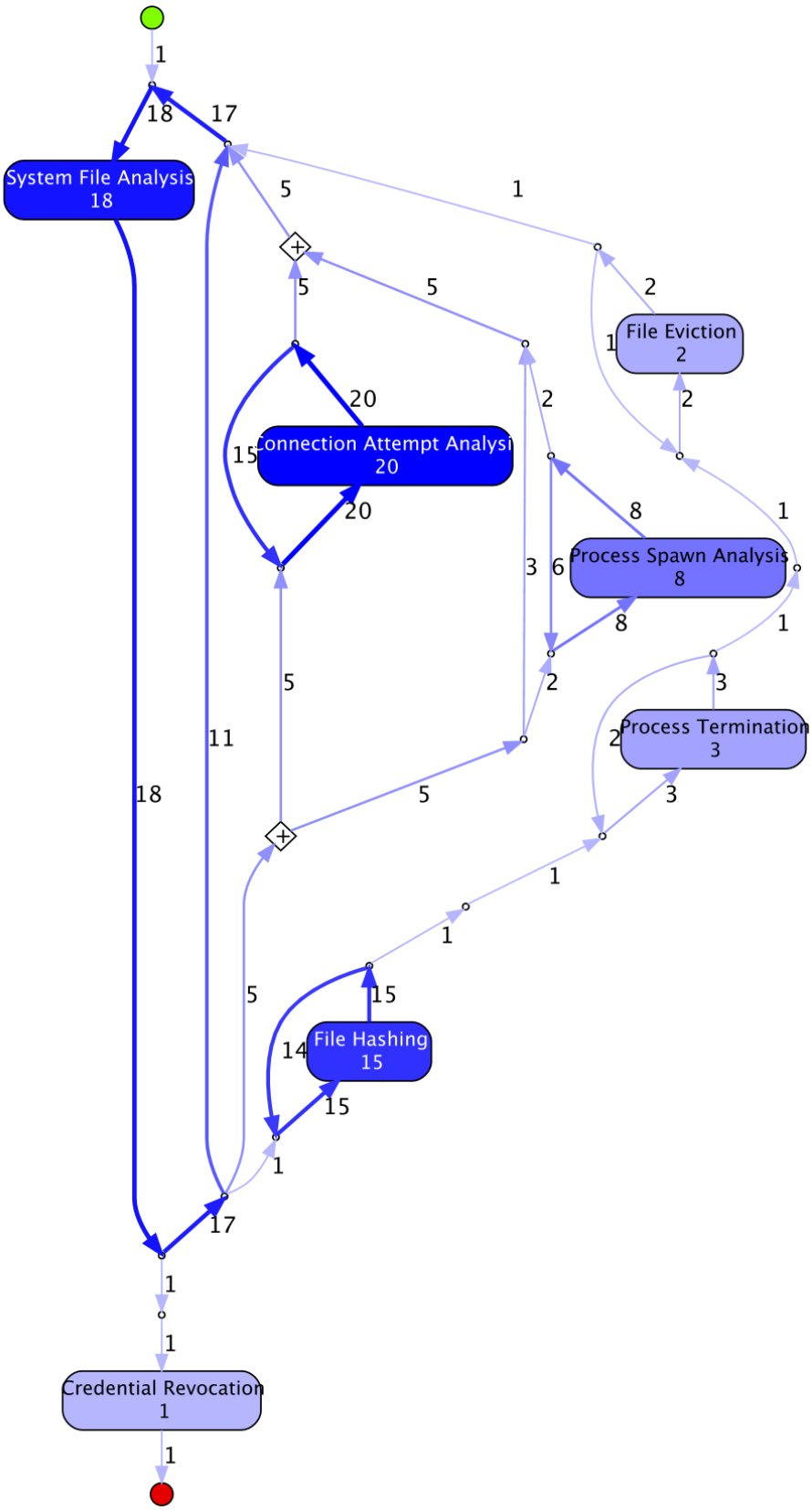



**Figura 5.11:** Escenario defensivo - Modelo del proceso de entrenamiento del grupo 6

# Capítulo 6

# Trabajos relacionados

En la literatura se pueden identificar varios trabajos que plantean distintas propuestas para la evaluación del entrenamiento de participantes en laboratorios de ciberseguridad. Se procede a presentar algunos de estos trabajos y se realiza una comparación con la metodología de evaluación definida en el Capítulo 4. La comparación será en base a cinco puntos fundamentales detallados a continuación.

### Eventos considerados

En primer lugar, para lograr evaluar el entrenamiento, es necesario monitorear y recolectar las acciones ejecutadas por los participantes de los entrenamientos. Si bien algunos trabajos tienen un enfoque más integral e incorporan en el análisis eventos clave que realizan los participantes y pueden estar representados por la ejecución de procesos, creación de archivos o conexiones de red; muchos de los trabajos analizados se concentran en el uso del historial de comandos ejecutados por los participantes de los entrenamientos.

Por ejemplo, un posible método para la evaluación de participantes en escenarios de ciberseguridad, a partir de los comandos ejecutados en una consola Linux, es propuesto en [81, 82]. Los autores recolectan el historial de comandos y en base a estos, generan grafos dirigidos que representan la actividad realizada por el participante, mostrando los comandos ejecutados y el orden de ejecución. Posteriormente, estos modelos visuales son analizados por instructores para determinar el cumplimiento de los objetivos de entrenamiento.

De forma similar, en [83] los autores proponen dos métodos para la evaluación basados en los comandos ejecutados por los participantes. En el primer

enfoque se construyen manualmente grafos de referencia que representan la actividad esperada en el escenario. Posteriormente se construyen los grafos de la actividad de los participantes, utilizando el historial de los comandos ejecutados, y se realiza la comparación con el grafo de referencia. Los nodos de estos grafos son subobjetivos que debe alcanzar el participante y las aristas dirigidas los comandos utilizados para alcanzarlos. En el segundo método se construyen lo que los autores denominan grafos de hitos. Este grafo contiene nodos de hitos o *templates nodes* que representan los subobjetivos del escenario y los comandos que deben ser ejecutados para alcanzarlos. Estos nodos están asociados en base a aristas dirigidas que denotan el orden en el que deben ser completados los hitos. Por otra parte, asociados a estos nodos de hitos, se tienen los nodos de intentos o *attempt nodes* que representan la actividad llevada a cabo por el estudiante para cumplir con el hito. Estos nodos registran la cantidad de intentos, tiempos y comandos utilizados en la ejecución.

Adicionalmente, en el trabajo [84], los mismos autores proponen una metodología enfocada en el uso de *pattern mining* y *clustering*. Nuevamente, la idea es a partir de comandos recolectados durante el entrenamiento de los participantes, esencialmente estos comandos son de dos tipos: comandos *bash* de una terminal Linux y comandos de la herramienta Metasploit [85]; aplicar estos dos enfoques para identificar comportamiento típico, dificultades y estrategias en común entre los participantes.

A su vez, existen trabajos que tienen una visión más general e incorporan en la evaluación, además de eventos recolectados durante el entrenamiento de los participantes, otro tipo de información recolectada mediante entrevistas y talleres mantenidos con los participantes de los entrenamientos. El trabajo presentado en [86] se realiza en el contexto del *cyber range* KYPO y trata la evaluación de escenarios complejos. En estos escenarios, los participantes se dividen en equipos y cada uno toma el rol de atacante (*red team*) o el rol de defensores (*blue team*). Adicionalmente, se tiene el *white team* conformado por los instructores y organizadores que definen los objetivos de entrenamiento y el contexto del ejercicio, y *green team* quienes brindan soporte a la infraestructura de entrenamiento. Los autores proponen que la evaluación de este tipo de ejercicio se realiza en base a análisis de las acciones y desempeño de los equipos durante el ejercicio, encuestas de retroalimentación y un taller de evaluación posterior. Exponen que este taller resulta muy útil para los participantes ya que pueden poner en común y comprender las acciones realizadas por los otros

equipos.

De forma similar al trabajo anterior, los autores Granasen et. al. proponen una metodología para la evaluación de escenarios complejos en los que participan múltiples equipos con roles distintos [87]. La metodología tiene en cuenta distintos aspectos para la evaluación como pueden ser: la performance del equipo (por ejemplo, cantidad de ataques exitosos realizados o ataques detectados), organización del equipo, comunicación entre miembros del equipo, observaciones realizadas por el *white team* sobre el actuar de un equipo e incluso tiene en consideración autoevaluaciones realizadas por los miembros de un equipo sobre su propio accionar.

En la metodología propuesta en nuestro trabajo planteamos el uso de todo tipo de eventos recolectados desde las máquinas donde se entrenan los participantes. Los múltiples eventos sucedidos en las máquinas, como pueden ser: procesos ejecutados, creación de archivos, conexiones de red establecidas, entre otros; son correlacionados según reglas del SIEM para identificar los eventos de interés o eventos clave en el entrenamiento. Este es un punto muy importante, ya que permite a los docentes reutilizar múltiples reglas de SIEM ya predefinidas (principalmente para escenarios de tipo ofensivo), facilitando así la tarea de evaluación. Adicionalmente, el uso de todos estos eventos recolectados por el SIEM permite una mayor visibilidad sobre las acciones realizadas por los participantes del entrenamiento, la cual no es posible tener analizando únicamente los comandos ejecutados. Por ejemplo, analizando los comandos no necesariamente se puede saber si su ejecución fue correcta o fallida, o en el caso que se ejecute un *script* no necesariamente se puede saber qué comandos se ejecutaron como parte de este.

### Recopilación de eventos

Como segundo punto, se analiza la forma en que la infraestructura de entrenamiento es monitoreada y cómo se obtienen los eventos, identificando dos grandes métodos. Por un lado, se tiene un monitoreo no invasivo en el cual los participantes de los entrenamientos no identifican que están siendo monitoreados y por lo tanto no pueden interferir con el monitoreo. Por el otro, un método invasivo que generalmente implica la instalación de un agente en la infraestructura de entrenamiento para la recolección de eventos. En este último caso, los participantes de los entrenamientos pueden identificar este servicio de

monitoreo y en algunos casos interferir con él.

En el trabajo [88], los autores proponen una forma de monitoreo y captura de evidencia *out-bound*, es decir, que los eventos recolectados de las máquinas virtuales donde se entrenan los participantes se capturan fuera de la propia máquina, interviniendo en la capa de virtualización. Esto permite que los participantes no identifiquen que están siendo monitoreados y no puedan interferir con este proceso. Lo que se captura es el estado de la memoria de las máquinas, el cual es analizado con el *framework* Volatility [89]. De este análisis se extraen eventos ocurridos en las máquinas, los cuales son correlacionados según una dependencia de objetos definida por los autores. Esta dependencia indica la relación entre eventos asociados a procesos, archivos y red. Posteriormente generan grafos, denominados *evidence graph*, que representan las acciones ejecutadas por los participantes y la dependencia entre estas acciones.

Los autores Maennel et. al. exponen una metodología de evaluación denominada "metodología de las cinco marcas de tiempo" [90]. La idea de la metodología es recopilar cinco *timestamps* o marcas de tiempo de acciones clave que suceden en ejercicios complejos de ciberseguridad en los cuales, en un mismo escenario, coexisten el *red team* y *blue team*. Estos *timestamps* son: inicio de ejecución de ataques por el *red team*, finalización (exitosa o fallida) de los ataques, detección de los ataques por parte del *blue team*, inicio de tareas de mitigación y contención del *blue team*, recuperación de los servicios por parte del *blue team*. Estos tiempos se recopilan de forma no intrusiva, analizando únicamente trazas de red que son recopiladas mientras ocurre el entrenamiento. A través de la medición de estos tiempos se podría identificar por ejemplo: cuánto tiempo necesitan los atacantes para comprometer la infraestructura, o cuánto tiempo necesitan los defensores para detectar dichos ataques y recuperarse de estos. Los autores indican que a medida que ocurran las sesiones de entrenamiento, es esperable que estos tiempos disminuyan, demostrando así el aprendizaje de los participantes.

De forma contraria, en el trabajo [91] los autores proponen un enfoque de monitoreo intrusivo en el contexto del *cyber range* THREAT-ARREST. Las máquinas en las que se entrenan los participantes cuentan con la instalación de un agente que recopila acciones ejecutadas por los participantes. Estas acciones son mapeadas a una jerarquía de categorías y subcategorías definidas por los autores. Básicamente, las categorías son: acciones sobre aplicaciones, acciones sobre sistemas operativos y acciones de red. Con estas acciones construyen

un *trainee graph*, es decir, un grafo que representa la actividad efectivamente realizada por el participante que se entrena en el *cyber range*. Asimismo, construyen un *reference graph*, o grafo de referencia, a partir de las trazas de ejecución ideales que son definidas por el instructor y provistas como parte del Cyber Threat and Training Preparation (CTTP) (el programa de entrenamiento que define cuestiones como la infraestructura del escenario). Ambos grafos son comparados a través de algoritmos definidos por los autores y, en base a esto, el participante obtiene un puntaje final sobre su desempeño en el entrenamiento.

Tectonic y su integración con Elastic Security permiten tanto la recolección de eventos de auditoría y seguridad de forma intrusiva a través de un agente instalado en las máquinas en las que se entrenan los participantes, como de forma no intrusiva a través de la recolección de eventos de red sin la necesidad de agentes instalados en las máquinas. Es importante aclarar que en este último caso, la visibilidad de las acciones que realizan los participantes es mucho menor, por lo cual los resultados obtenidos con la aplicación de la metodología de evaluación pueden carecer de cierto nivel de detalle y no ser lo suficientemente enriquecedores para la evaluación demandada por el docente.

### Procesamiento y análisis de eventos

Una vez que se tienen los datos recolectados, los distintos trabajos proponen algún tipo de procesamiento y análisis para evaluar el entrenamiento. Muchos de los trabajos plantean el uso de grafos para la representación visual de las acciones ejecutadas por los participantes y su orden de ejecución. Y de estos, varios proponen la comparación de estos grafos con un grafo de referencia que especifica la actividad ideal que debería ser realizada para cumplir con el entrenamiento. La comparación de estos grafos permitiría identificar el cumplimiento de los objetivos de entrenamiento o casos de trampa. Varios de estos trabajos utilizan algoritmos de grafos o proponen algoritmos *ad hoc*.

Por ejemplo, los autores Andreolini et al. hacen una propuesta para la evaluación inspirada en modelos de ataque [92]. Un modelo de ataque permite representar la actividad realizada por un actor malicioso a la hora de explotar vulnerabilidades de un sistema para cumplir con cierto objetivo. La propuesta de los autores es modelar la actividad de los participantes utilizando grafos dirigidos, donde los vértices o nodos representan estados por los que pasa el

participante durante el entrenamiento y las aristas, las actividades realizadas para pasar de un estado a otro. Utilizando estos grafos dirigidos, construyen dos modelos que en su trabajo los denominan: *reference graph* y *trainee graph*. El primero es construido por un instructor de forma manual y representa el comportamiento ideal o la actividad esperada a realizar para cumplir con los objetivos de enseñanza del escenario. El segundo modelo, que es construido de forma automática por el *framework* que desarrollaron en base a los eventos recolectados durante el entrenamiento, representa la actividad real llevada a cabo en el escenario por los participantes. Como parte del trabajo, también proponen distintos algoritmos que permiten comparar el modelo de referencia y el modelo del participante, generando un puntaje final de performance que indica qué tan bien se entrenó el participante.

A su vez, dichos autores extienden su trabajo planteando mejoras para la evaluación de escenarios más complejos [93]. Si bien los grafos construidos y los algoritmos diseñados para la comparación de dichos grafos funcionan muy bien en escenarios sencillos, alegan que en escenarios complejos donde la cantidad de actividades es muy grande, y por lo tanto la cantidad de nodos y aristas de estos grafos aumenta, sus algoritmos no funcionan con la misma eficacia. Por lo tanto, su propuesta de mejora es reducir la complejidad de los grafos que se utilizan como insumo para los algoritmos de comparación. La idea es representar los ejercicios a través de grafos locales más pequeños, que detallan las actividades realizadas por los participantes para cumplir con desafíos intermedios, y un grafo global más simple que representa las conexiones entre dichos desafíos. Al introducir los grafos globales y locales, ahora es posible evaluar por separado distintas partes del escenario, que están representadas por los grafos locales, asignando un puntaje distinto según un mayor peso que le quiera dar el instructor a cierta parte del escenario (por ejemplo, por presentar una mayor complejidad o resultar más importante en el contexto del escenario completo).

De forma similar, en los trabajos [81, 82, 83, 91] ya presentados, también se propone el uso de grafos para modelar el entrenamiento de los participantes.

En cambio, la metodología propuesta en nuestro trabajo utiliza como pilar fundamental minería de procesos dado que el entrenamiento de los participantes en un *cyber range* puede ser entendido como un proceso. Solo uno de los casos analizados aplica minería de procesos. Este es el trabajo de Macak et al. [94, 95] que investigan sobre la aplicación de minería de procesos en la

evaluación de la actividad de participantes en el *cyber range* KYPO. Plantean la construcción de logs de eventos a partir de tres tipos o categorías de información:

- *Game Events*: son eventos generados por la interacción del participante con la interfaz de KYPO. Por ejemplo, estos eventos pueden ser generados cuando el participante solicita una pista o entrega una *flag*.
- *Bash history*: historial de todos los comandos ejecutados en una terminal o consola Linux de una instancia que forma parte del escenario.
- *Metasploit commands*: son todos los comandos ejecutados que están asociados al uso de la herramienta Metasploit.

Cada evento que conforma el log de eventos consta de los siguientes campos: tiempo en el que se ejecutó el evento, tipo de evento (se corresponde a uno de los tres tipos presentados anteriormente), evento (representa una actividad concreta realizada por el participante, es una subcategorización dentro del tipo de evento), aprendiz (identificador del participante que realizó el evento) y parámetros opcionales del evento. Teniendo en cuenta estos campos, es bastante natural que el campo evento se utilice como la actividad, el tiempo como *timestamp* y el aprendiz como identificador del caso. Utilizando estos logs de eventos, los autores aplican descubrimiento de proceso para generar modelos que representan la actividad de los aprendices dentro del escenario.

Sin embargo, dicho trabajo se centra fuertemente en la aplicación de descubrimiento de procesos. En cambio, la metodología introducida en nuestro trabajo hace uso de todas las herramientas de minería de procesos, incluyendo descubrimiento de procesos, verificación de conformidad y mejora de procesos; brindando así una herramienta integral que permite el estudio y análisis del entrenamiento de los participantes en base al proceso conformado por las actividades de ataque y defensa que aplican como parte de su entrenamiento.

Gracias al uso de todas las herramientas brindadas por PM, es que es posible el descubrimiento y mejora de modelos de referencia que representan la actividad esperada de entrenamiento y que posteriormente pueden ser utilizados como referencia para la comparación con la actividad realizada por los participantes en su entrenamiento. Generalmente, los trabajos que plantean el uso de algún tipo de modelo o grafo de referencia asumen que son construidos por el instructor que especifica el escenario, como en los casos de los trabajos [91, 92]. Esto puede llegar a ser una tarea ardua para escenarios de un

mayor nivel de complejidad, e incluso en aquellos casos en los que es posible especificar modelos de referencia, pueden quedar obsoletos rápidamente a raíz del propio avance de la tecnología, descubrimiento de nuevas vulnerabilidades y surgimiento de nuevas técnicas y herramientas de ataque y defensa que no fueron visualizadas e incluidas en el diseño del escenario. Por tal razón, nuestra metodología tiene en cuenta que los modelos de referencia puedan ser generados en base a la actividad que aplican los participantes en los entrenamientos y mejorados con conocimiento que incorpora el instructor como experto en ciberseguridad y diseñador del escenario, facilitando así la tarea de creación de dichos modelos.

Otro punto clave de diferenciación de la metodología propuesta en nuestro trabajo es el uso de los *frameworks* MITRE ATT&CK y MITRE D3FEND para la especificación y comunicación de las actividades realizadas por los participantes en el entrenamiento. Tanto el log de eventos como modelos generados quedan expresados en base a tácticas y técnicas de estos *frameworks*, lo cual implica que se encuentran expresados en base a términos comunes y adoptados ampliamente por la comunidad de ciberseguridad, facilitando así su comprensión. Adicionalmente, mediante el uso de estos *frameworks* es posible generar modelos con distinto nivel de granularidad o con distintos niveles de abstracción. Los modelos pueden ser generados a nivel de las tácticas o las técnicas. El primer caso permite un análisis en base a objetivos de alto nivel, mientras que el segundo en base a tareas específicas para cumplir con dichos objetivos, y por lo tanto, permite comprender de mejor forma cómo se entrena el participante.

#### Resultados generados

Otro de los aspectos a considerar puede ser el tipo de resultado obtenido a raíz de la aplicación de las distintas metodologías de evaluación. Algunos de los trabajos se centran en obtener como resultado un puntaje que representa qué tan bien resultó el entrenamiento de un participante (o un equipo de participantes en caso de escenarios más complejos), generalmente utilizando funciones de puntaje basadas en distintas cuestiones asociadas al entrenamiento como pueden ser eventos ejecutados, tiempos de ejecución, número de intentos, pistas solicitadas, entre otras.

En los trabajos [96, 97] se presenta una herramienta que permitiría la eva-

luación del entrenamiento aplicando ideas de la gamificación y se podría integrar a distintos *cyber ranges*. Esta herramienta recibe mensajes desde los *cyber ranges* y en base a esto calcula un puntaje sobre la performance de los participantes en su entrenamiento. Los mensajes recibidos desde el *cyber range* deben incluir:

- Objetivos: describen las metas del ejercicio.
- Acciones: identifican los pasos realizados por los participantes para cumplir con el objetivo. Las acciones pueden ser de tres tipos: *flag* (indicando que el participante encontró y entregó la bandera), tarea (identifica una tarea que el participante debe completar) y ataque (un ataque que el participante debe ejecutar para comprometer cierto servicio o sistema de la infraestructura).
- Pistas: las acciones tienen asociadas pistas que los participantes pueden solicitar como ayuda.
- Tiempo: especifica el momento exacto en el que se ejecutó cada acción y se compara respecto a un tiempo óptimo definido por el instructor (cuando diseña el escenario).
- Peso: permite darle una importancia a cada acción ejecutada.
- Nivel de dificultad: asociado a cada acción ejecutada.
- Resultado: identifica si cada acción ejecutada resultó exitosa o no.

Utilizando estos mensajes, la herramienta asigna un puntaje al entrenamiento del participante, aplicando un algoritmo definido por los autores del trabajo. Este algoritmo tiene en consideración los distintos componentes del mensaje, como puede ser: comparación del tiempo respecto al tiempo óptimo, dificultad de la tarea (a mayor dificultad más puntaje), si se solicitaron pistas (cuantas más pistas fueron solicitadas menor puntaje se asigna), entre otras cuestiones.

En el trabajo [98], los autores proponen una metodología similar a la presentada anteriormente para medir Situational Awareness (SA) de los participantes del entrenamiento. SA hace referencia al nivel de entendimiento de un participante del entrenamiento sobre la situación actual, la infraestructura y cómo sus acciones pueden afectar el cumplimiento de los objetivos de entrenamiento. La idea del trabajo es que en base a mensajes obtenidos desde el *cyber range*, cuyo contenido es muy similar al presentado en el trabajo anterior, aplican una función de puntaje que calcula el puntaje final del entrenamiento de los participantes y los equipos participantes.

En principio, si bien las técnicas de verificación de conformidad pueden dar métricas numéricas como resultado de la comparación de un modelo de proceso y un log de eventos, la metodología que proponemos no tiene como foco obtener un puntaje que represente la performance del participante en su entrenamiento. Por el contrario, la idea con esta metodología es brindar herramientas que sirvan de apoyo para la evaluación que realiza el docente y como resultado primordial obtener modelos visuales que permiten representar y explicar la actividad llevada a cabo por los participantes en un lenguaje común y de fácil entendimiento. En otras palabras, el docente puede aplicar la metodología y utilizar los resultados como insumos para la evaluación y posterior *feedback* que puede brindar al participante.

Es importante destacar que la metodología propuesta en nuestro trabajo posibilita el análisis del entrenamiento de participantes de un *cyber range* en base a distintas perspectivas. Por un lado, el análisis se puede centrar en las tareas aplicadas y orden de ejecución, como es el caso de la mayoría de los trabajos analizados. Sin embargo, el análisis también puede centrarse en la perspectiva de tiempo, identificando así tareas más costosas dado que los participantes demoran más tiempo en lograr ejecutarlas o, de forma contraria, tareas más sencillas. En este sentido, son pocos los trabajos analizados que permiten el análisis en base a esta perspectiva.

### Objetivos de aplicación

La gran mayoría de trabajos analizados tienen como fin primordial la evaluación del entrenamiento de participantes de talleres prácticos de ciberseguridad. Sin embargo, el autor Kakouros presenta un trabajo con un foco distinto. El autor propone un *framework* para la detección de trampas en el contexto de un *cyber range* [99]. Según el autor, los escenarios pueden representarse en base a dos grafos dirigidos acíclicos, el primero registra las rutas de la red del escenario y el segundo es el grafo de ataque del escenario. Este último se compone de la siguiente forma: los nodos del grafo son las instancias o máquinas que conforman un escenario y las aristas representan las dependencias de resolución entre dos instancias. Cada nodo del grafo (instancia) tiene asociado un conjunto de *flags* (por lo general suelen ser archivos de texto alojados en el sistema de archivos de las instancias) que son los elementos que debe capturar el participante y entregar para que se pueda dar por resuelta la instancia. A su

vez, cada *flag* tiene asociado un conjunto de actividades mínimas que el participante debe realizar para lograr obtenerlas, y un conjunto de pistas que el participante puede solicitar como forma de ayuda. Por último, cada pista tiene asociado un conjunto de canarios, que identifican de forma única a una pista, y se utilizan para detectar trampas. En caso de que un participante reutilice un canario, el *framework* lo catalogará como tramposo. Por otra parte, el *framework* recolecta información (eventos de red, procesos, archivos, solicitud de pistas, entre otros) de las actividades llevadas a cabo por el participante durante la práctica. Utilizando la información recolectada de la ejecución del participante y los grafos que representan al escenario, el *framework* intenta verificar si el participante reutilizó canarios o entregó una *flag* habiendo omitido alguna de las actividades mínimas necesarias para obtenerlas. En estos casos se considera que el participante ha hecho trampa.

La metodología que nosotros proponemos también puede ser utilizada para la detección de trampa, siguiendo una idea muy similar planteada al trabajo presentado anteriormente. Mediante la aplicación de minería de procesos, y en particular *conformance checking* es posible identificar desviaciones entre los modelos generados a raíz de eventos de entrenamientos de los participantes y modelos que representan la actividad mínima necesaria para completar el escenario. En caso de detectarse desviaciones entre estos modelos, y en particular en caso de que el participante entregue las *flags* sin haber realizado uno de los pasos necesarios para cumplir con el escenario, se pueden identificar potenciales casos de trampa.

Por último, como ya fue presentado, nuestra metodología también puede ser empleada para la evaluación del entrenamiento de participantes de un *cyber range* y para el análisis del diseño de los escenarios de dichas plataformas. En definitiva, esta metodología brinda al instructor la posibilidad de evaluar sesiones de entrenamiento en un *cyber range* e identificar posibles brechas y áreas de mejora, adquiriendo así una comprensión más profunda de los procesos involucrados en la realización de estas actividades de capacitación.

# Capítulo 7

# Conclusiones

En este trabajo presentamos una metodología para la evaluación del entrenamiento de usuarios de un *cyber range*, utilizando como pilares fundamentales: una herramienta de SIEM para el análisis y correlación en vivo de los eventos ejecutados por los usuarios en la infraestructura de entrenamiento, los *frameworks* MITRE ATT&CK y MITRE D3FEND para la categorización y expresión de estos eventos en términos comunes y ampliamente adoptados por la comunidad de ciberseguridad, y la minería de procesos para la generación de modelos que representan la actividad llevada a cabo por los estudiantes y adicionalmente su comparación con modelos de referencia que expresan la actividad ideal que debe ser realizada para el cumplimiento de los objetivos de entrenamiento de un escenario.

A su vez, para lograr aplicar la metodología en el *cyber range* Tectonic, debimos integrar la plataforma con dos herramientas. Por un lado, la herramienta de SIEM Elastic Security para el monitoreo de la actividad de los participantes de los entrenamientos y posterior análisis y categorización según los *frameworks* de MITRE en base a analíticas de seguridad. Nuestra propuesta incluye utilizar esta herramienta de forma innovadora, para permitir además del estudio de actividad ofensiva, el estudio de actividad defensiva. Por otro lado, también integramos la herramienta Caldera para la implementación de ataques automatizados sobre la infraestructura de entrenamiento lo cual posibilita el dictado de escenarios de tipo *blue team*.

Nuestra metodología permite la evaluación tanto de entrenamientos *red team* como *blue team*, es decir, que es posible evaluar escenarios en los que el usuario toma el rol de atacante o de defensor, cubriendo así ambos aspectos

del entrenamiento en ciberseguridad. La metodología dota al instructor de herramientas que le permiten identificar no solamente si los usuarios cumplen los objetivos de entrenamiento, sino que, más importante aún, cómo lo hacen. Es posible que el instructor obtenga detalles de las herramientas y técnicas empleadas por los participantes del entrenamiento.

Adicionalmente al análisis del entrenamiento de los usuarios, esta metodología puede ser empleada para la detección de trampas al comparar la actividad registrada de un usuario con un modelo de referencia que detalle las actividades mínimas necesarias para el cumplimiento de un escenario. En caso de detectar desviaciones y que el usuario haya hecho entrega de las *flags* se podría llegar a la conclusión de que el usuario ha cometido trampa. Por otra parte, la metodología también podría ser utilizada para la identificación de errores de diseño del escenario, ya que la ocurrencia de desviaciones respecto al modelo de referencia puede deberse a que el usuario ha identificado un nuevo camino de resolución que no había sido considerado por el instructor a la hora de generar el escenario y que por lo tanto "rompe" el flujo de ejecución ideado.

Mediante el estudio del arte realizado sobre trabajos relacionados del área, se identificaron algunos puntos de innovación de la metodología presentada. La metodología hace uso de los *frameworks* MITRE ATT&CK y MITRE D3FEND para la categorización de la actividad ejecutada por los participantes de los entrenamientos en tácticas y técnicas empleadas por atacantes y defensores. Esto permite reutilizar todo el conocimiento embebido en estos *frameworks* y expresar la actividad de los participantes en un lenguaje común que simplifica la comprensión por parte del instructor. Adicionalmente, brinda la posibilidad de analizar la actividad de los participantes en distintos niveles de granularidad, siendo posible expresar su actividad y generar modelos a niveles de tácticas, que representan los objetivos a alto nivel, o a nivel de técnicas, las cuales permiten entender con mayor detalle cómo cumplen con esos objetivos. Por otra parte, mediante la aplicación de la metodología es posible descubrir modelos de referencia que detallen la actividad ideal de entrenamiento en base a la propia actividad ejecutada por los participantes, incluso en escenarios complejos donde esta tarea podría ser muy dificultosa para que un instructor la realice de forma manual. Estos modelos de referencia pueden ser complementados con el conocimiento que aporta el instructor que diseña el escenario, generando así modelos de referencia más representativos y confiables.

En lo que refiere a trabajo a futuro, se tienen identificadas distintas líneas

de investigación para la extensión de este trabajo. En primer lugar, en base a los casos de aplicación se identificaron problemáticas asociadas al uso de las reglas de SIEM que permiten la detección de actividades claves realizadas por los participantes del entrenamiento y su categorización según los *frameworks* de MITRE. En particular se identificaron la existencia de falsos positivos, alertas que no deberían haber sido generadas, y de falsos negativos, alertas que deberían haber sido generadas.

El uso de un conjunto de reglas de SIEM completo y correcto es de vital importancia en la metodología para la identificación y categorización de los eventos de interés. Por lo tanto, se plantea complementar las reglas estáticas utilizadas, generalmente basadas en umbrales, con reglas que hagan uso de herramientas del dominio de *machine learning* lo que puede ayudar a mejorar la detección de las actividades claves realizadas por los usuarios en su entrenamiento.

Asociadas a las analíticas de SIEM, si bien existen múltiples repositorios de reglas para la detección de actividad maliciosa, se identificó que no son nada comunes reglas que permitan detectar y categorizar actividad defensiva. Esto dificulta la implementación de la metodología en escenarios de tipo *blue team*, siendo de vital importancia el conocimiento del instructor para la generación de estas reglas. Por lo tanto, otra línea de trabajo a futuro podría ser generar un catálogo de analíticas o reglas de SIEM para la detección de actividad defensiva que se encuentren caracterizadas según las tácticas y técnicas del *framework* MITRE D3FEND.

Por otra parte, se mostró que es posible aplicar la metodología y en particular generar modelos del entrenamiento de los participantes en base a tácticas o técnicas de los *frameworks* de MITRE. Podría ser interesante, basándose en la idea planteada por uno de los trabajos del área presentados, analizar la posibilidad de generar dos tipos de modelos: modelos globales más simples construidos a partir de las tácticas, y modelos locales, centrados en cierta porción del entrenamiento, generados en base a las técnicas. Esto podría resultar muy útil en escenarios complejos donde el análisis de un único modelo basado en técnicas se puede tornar dificultoso por la cantidad de actividades y caminos distintos representados en el modelo. De esta forma, un instructor podría lograr un análisis más enriquecedor ya que en primera instancia contaría con un modelo global simplificado que muestre las acciones ejecutadas por los participantes en alto nivel, y luego podría obtener un mayor nivel de detalle de

cómo se entrenan los participantes en cierta porción del escenario al contar con modelos locales que expresan las actividades en base a las técnicas.

En el caso de estudio del escenario ofensivo se observó que la inmensa mayoría de las actividades que conforman el log de eventos están asociadas al acceso a credenciales. Para lograr obtener dichas credenciales, los participantes utilizan una herramienta automática de ataque que realiza múltiples intentos de autenticación con un servicio hasta obtener credenciales válidas. En definitiva, el uso de herramientas automáticas podría generar una gran cantidad de eventos que "inunden" el log de eventos, dando la sensación de que el participante concentró sus esfuerzos en cierta tarea cuando realmente su esfuerzo, conocimiento y habilidades fueron aplicadas en otras tareas de un mayor grado de complejidad pero que se encuentran evidenciadas con una menor frecuencia en el log de eventos y modelos. Por lo tanto, para cierto tipo de análisis que desee realizar el docente podría ser interesante generar modelos que de cierta forma tengan en cuenta el uso de herramientas automáticas, agrupando los múltiples eventos que generan estas herramientas para que sean representados una única vez. Esto se podría hacer a nivel de log de eventos una vez que se tienen todos los eventos de interés generados o se podría hacer directamente en las reglas de SIEM utilizando reglas con umbrales más altos.

Es importante destacar que para otro tipo de análisis, al instructor le puede resultar interesante cómo los participantes hacen uso de las herramientas automáticas, por lo que también es importante poder generar modelos que permitan representar todas las tareas asociadas al uso de dichas herramientas. Por ejemplo, en el caso de estudio del escenario ofensivo, hubo participantes que ejecutaron ataques de fuerza bruta para la obtención de credenciales, pero estos ataques no fueron evidenciados por las reglas de SIEM utilizadas o fueron evidenciados en una menor cantidad de veces. Se podría pensar que estos participantes fueron más cuidadosos a la hora de ejecutar los ataques generando un menor "ruido" y este puede ser un punto importante a la hora de evaluar sus habilidades como atacante, ya que demuestra que aprendieron cómo utilizar y configurar la herramienta con la intención de que sus ataques no fueran detectados.

Por último, otra propuesta es extender la metodología para ser aplicada con el objetivo de apoyar el dictado de entrenamientos adaptativos. En este enfoque la idea es que los escenarios de entrenamiento sean modificados de forma dinámica según los avances de los participantes del entrenamiento, lo-

grando así que el entrenamiento sea adaptado al conocimiento y habilidades de cada participante y por lo tanto maximizando el beneficio que este obtiene del entrenamiento. Para esto, es necesario extender el lenguaje de definición de escenarios y las capacidades de Tectonic para que los escenarios puedan ser modificados o evolucionen a medida que avanza el entrenamiento. En lo que refiere a los escenarios, podrían ser planteados de forma tal que existan distintos caminos de resolución, de distinto nivel de dificultad, para alcanzar un objetivo único. A medida que el participante se entrena, se podría monitorear su actividad en tiempo real y estudiar el proceso generado a raíz de dicha actividad. Para el estudio en tiempo real del proceso asociado al entrenamiento de los usuarios podría ser útil aplicar *streaming process mining* [100].

Al comparar la actividad realizada hasta el momento por el participante del entrenamiento con un modelo de referencia, se podría conocer en que etapa del entrenamiento se encuentra y en particular si está atrasado o adelantado según lo esperado. Si el participante demuestra un rendimiento superior al esperado se puede implementar una transformación del escenario para obligarlo a tomar un camino de resolución de mayor dificultad. Por ejemplo, en un escenario de tipo defensivos se podrían lanzar ataques en tiempo real utilizando Caldera lo que requiere que el participante reaccione ante estos ataques y defienda la infraestructura. En escenarios ofensivos se podrían aplicar técnicas del dominio Moving Tagret Defense (MTD) [101] para implementar controles o medidas de seguridad que dificulten las tareas de ataque del participante. Por el contrario, si el participante demuestra un retraso en su entrenamiento, esto podría ser evidenciado a tiempo y el instructor podría brindar *feedback* inmediato para ayudar al participante en su avance. En definitiva, esta mejora permitiría extender la metodología y mejorar el propio *cyber range* para brindar escenarios adaptativos que puedan ser ajustados de forma dinámica en función de las capacidades y habilidades de los usuarios, logrando así que estos obtengan una experiencia más enriquecedora de estas sesiones de entrenamiento.

# Referencias bibliográficas


[1] Steve Morgan. *Cybercrime To Cost The World $10.5 Trillion Annually By 2025*. En línea, último acceso: 24/04/2025. 2020. URL: https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/.

[2] CERTuy. *En 2024 aumentó la detección de incidentes de seguridad de la información*. En línea, último acceso: 24/04/2025. 2025. URL: https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/comunicacion/noticias/2024-aumento-deteccion-incidentes-seguridad-informacion.

[3] Steve Morgan. *Cybersecurity Jobs Report: 3.5 Million Unfilled Positions In 2025*. En línea, último acceso: 24/04/2025. 2023. URL: https://cybersecurityventures.com/jobs/.

[4] *Tectonic*. https://www.fing.edu.uy/inco/proyectos/tectonic. En línea, último acceso: 24/04/2025.

[5] Guillermo Guerrero, Gustavo Betarte and Juan Diego Campo. "Process Mining-Based Assessment of Cyber Range Trainings". En: *2024 L Latin American Computer Conference (CLEI)*. 2024, pp. 1-10. DOI: 10.1109/CLEI64178.2024.10700452.

[6] Guillermo Guerrero, Gustavo Betarte and Juan Diego Campo. "Tectonic: An Academic Cyber Range". En: *2024 IEEE Biennial Congress of Argentina (ARGENCON)*. 2024, pp. 1-8. DOI: 10.1109/ARGENCON62399.2024.10735713.

[7] Guillermo Guerrero, Gustavo Betarte and Juan Diego Campo. "Assessment of Red and Blue Team Training in the Cyber Range Tectonic using Process Mining". unpublished.

[8] National Initiative for Cybersecurity Education (NICE) - Cyber Range Project Team. *The Cyber Range: A Guide. Guidance Document for the Use Cases, Features, and Types of Cyber Ranges in Cybersecurity Education, Certification and Training*. https://www.nist.gov/system/files/documents/2023/09/29/The%20Cyber%20Range_A%20Guide.pdf. En línea, último acceso: 24/04/2025. 2023.

[9] April C. Wright. "Orange Is The New Purple". En: *BlackHat USA* (2017).

[10] Louis Cremen. *Introducing the InfoSec colour wheel, blending developers with red and blue security teams*. https://hackernoon.com/introducing-the-infosec-colour-wheel-blending-developers-with-red-and-blue-security-teams-6437c1a07700. En línea, último acceso: 24/04/2025. 2018.

[11] R. Flinterman et al. *Clusus: a cyber range for network attack simulations*. Delft, Países Bajos, 2019.

[12] Cyber Range Organization and Design (CROND). Japan Advanced Institute of Science and Technology. *CyTrONE User Guide*. Inf. téc. En línea, último acceso: 24/04/2025. 2021.

[13] Jan Vykopal et al. "Scalable Learning Environments for Teaching Cybersecurity Hands-on". En: *2021 IEEE Frontiers in Education Conference (FIE)*. 2021, pp. 1-9. DOI: 10.1109/FIE49875.2021.9637180.

[14] George Hatzivasilis et al. "The THREAT-ARREST Cyber Range Platform". En: *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. 2021, pp. 422-427. DOI: 10.1109/CSR51186.2021.9527963.

[15] *Cyberwiser.ew*. https://cyberwiser.eu. En línea, último acceso: 24/04/2025.

[16] *Cyberbit*. https://www.cyberbit.com. En línea, último acceso: 24/04/2025.

[17] *TryHackMe*. https://tryhackme.com. En línea, último acceso: 24/04/2025.

[18] *Hack The Box*. https://www.hackthebox.com. En línea, último acceso: 24/04/2025.

[19] *YAML*. https://yaml.org. En línea, último acceso: 24/04/2025.

[20] RedHat. *Ansible*. https://www.ansible.com/. En línea, último acceso: 24/04/2025.

[21] Ansible. *Ansible Community Documentation - Glossary*. https://docs.ansible.com/ansible/latest/reference_appendices/glossary.html. En línea, último acceso: 21/06/2025.

[22] *Docker*. https://www.docker.com. En línea, último acceso: 24/04/2025.

[23] Oracle. *Virtualbox: Powerful open source virtualization*. https://www.virtualbox.org. Online, last accessed: 29/03/2025.

[24] Cuong Pham et al. "CyRIS: a cyber range instantiation system for facilitating security training". En: *Proceedings of the Seventh Symposium on Information and Communication Technology*. 2016, pp. 251-258.

[25] *Libvirt*. https://libvirt.org. En línea, último acceso: 24/04/2025.

[26] *Amazon Web Services*. https://aws.amazon.com/. En línea, último acceso: 24/04/2025.

[27] HasiCorp. *Packer*. https://www.packer.io/. En línea, último acceso: 24/04/2025.

[28] OpenStack. *Open Source Cloud Computing Infrastructure*. https://www.openstack.org. En línea, último acceso: 24/04/2025.

[29] OpenNebula. *The Open Source Cloud Edge Computing Platform*. https://opennebula.io. Online, last accessed: 29/03/2025.

[30] Razvan Beuran et al. "Realistic Cybersecurity Training via Scenario Progression Management". En: *IEEE European Symposium on Security and Privacy Workshops* (2019), pp. 67-76.

[31] R. Beuran et al. "Integrated framework for hands-on cybersecurity training: CyTrONE". En: *ScienceDirect Computer and Security* 78 (2018), pp. 43-59.

[32] Moodle. *Moodle Documentation*. https://docs.moodle.org/. En línea, último acceso: 24/04/2025.

[33] Wil van der Aalst. *Process Mining Data Science in Action*. Springer Berlin, Heidelberg, 2016.

[34] W. van der Aalst, T. Weijters and L. Maruster. "Workflow mining: discovering process models from event logs". En: *IEEE Transactions on Knowledge and Data Engineering* 16.9 (2004), pp. 1128-1142. DOI: 10.1109/TKDE.2004.47.

[35] T. Weijters, W. van der Aalst and Alves Medeiros Alves. *Process Mining with the Heuristics Miner-algorithm*. Vol. 166. Enero de 2006.

[36] Sander J. J. Leemans, Dirk Fahland and Wil van der Aalst. "Discovering Block-Structured Process Models from Event Logs - A Constructive Approach". En: *Application and Theory of Petri Nets and Concurrency*. Ed. por José-Manuel ColomJörg Desel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 311-329. ISBN: 978-3-642-38697-8.

[37] Jörg Desel, Wolfgang Reisig and Grzegorz Rozenberg. *Lectures on Concurrency and Petri Nets: Advances in Petri Nets*. 2004.

[38] Remco M. Dijkman, Marlon Dumas and Chun Ouyang. "Semantics and analysis of business process models in BPMN". En: *Information and Software Technology* 50.12 (2008), pp. 1281-1294. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2008.02.006. URL: https://www.sciencedirect.com/science/article/pii/S0950584908000323.

[39] Wil van der Aalst, Joos Buijs and Boudewijn van Dongen. "Towards Improving the Representational Bias of Process Mining". En: *Data-Driven Process Discovery and Analysis*. Ed. por Karl Aberer, Ernesto DamianiTharam Dillon. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 39-54. ISBN: 978-3-642-34044-4.

[40] A. Rozinat and W. van der Aalst. "Conformance checking of processes based on monitoring real behavior". En: *Information Systems* 33.1 (2008), pp. 64-95. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2007.07.001. URL: https://www.sciencedirect.com/science/article/pii/S030643790700049X.

[41] Wil Aalst, Arya Adriansyah and Boudewijn Dongen. "Replaying History on Process Models for Conformance Checking and Performance Analysis". En: *WIREs Data Mining and Knowledge Discovery* 2 (marzo de 2012), pp. 182-192. DOI: 10.1002/widm.1045.

[42] *MITRE*. https://www.mitre.org. En línea, último acceso: 24/04/2025.

[43] MITRE. *D3FEND*. https://d3fend.mitre.org. Online, last accessed 12/04/2025.

[44] Blake E. Strom et al. *MITRE ATT&CK: Design and Philosophy*. 2018.

[45] Peter E. Kaloroumakis and Michael J. Smith. "Toward a knowledge graph of cybersecurity countermeasures". En: *The MITRE Corporation* 11 (2021).

[46] Juan Diego Campo, Lucía Escanellas and Carlos Pintado. *Design and Development of a Framework For IT-security Training*. Montevideo, Uruguay, 2009.

[47] *Red Ciberlac*. https://ciberlac.org. En línea, último acceso: 24/04/2025.

[48] Elastic. *Elastic Security*. https://www.elastic.co/security. En línea, último acceso: 24/04/2025.

[49] Rodrigo Gallardo and Guillermo Guerrero. *Reingeniería del Laboratorio de Seguridad Informática análisis, diseño e implementación de un Cyber Range*. Available at https://hdl.handle.net/20.500.12008/30925. Montevideo, Uruguay, 2021.

[50] Elastic. *Elastic Docs - Prevent Elastic Agent uninstallation*. https://www.elastic.co/docs/solutions/security/configure-elastic-defend/prevent-elastic-agent-uninstallation. En línea, último acceso: 21/06/2025.

[51] AWS. *What is Traffic Mirroring?* https://docs.aws.amazon.com/vpc/latest/mirroring/what-is-traffic-mirroring.html. Online, last accessed: 29/03/2025.

[52] Gabriel Corujo and Manuel Rodriguez. *CyRa.uy: Hacia un cyber range académico*. https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/42955. Montevideo, Uruguay, 2023.

[53] Mitre. *Caldera*. https://caldera.mitre.org. En línea, último acceso: 24/04/2025.

[54] Eoghan Casey. *Handbook of Digital Forensics and Investigation*. USA: Academic Press, Inc., 2009. ISBN: 0123742676.

[55] *Azure*. https://azure.microsoft.com. En línea, último acceso: 24/04/2025.

[56] *Google Cloud Platform*. https://cloud.google.com. En línea, último acceso: 24/04/2025.

[57] John Klein and Douglas Reynolds. *Infraestructure as Code: Final Report*. Inf. téc. Software Engineering Institute, Carnegie Mellon University, diciembre de 2018.

[58] HasiCorp. *Terraform*. https://www.terraform.io/. En línea, último acceso: 24/04/2025.

[59] Python.org. *Welcom to Python.org*. https://www.python.org. En línea, último acceso: 24/04/2025.

[60] Elastic. *Elastic Stack*. https://www.elastic.co/elastic-stack. En línea, último acceso: 24/04/2025.

[61] Chris Eagle. "Computer Security Competitions: Expanding Educational Outcomes". En: *IEEE Security Privacy* 11.4 (2013), pp. 69-71. DOI: 10.1109/MSP.2013.83.

[62] Sander J.J. Leemans. "Robust Process Mining with Guarantees". En: *International Conference on Business Process Management*. 2017. URL: https://api.semanticscholar.org/CorpusID:52196338.

[63] B. F. van Dongen et al. "The ProM Framework: A New Era in Process Mining Tool Support". En: *Applications and Theory of Petri Nets 2005*. Ed. por Gianfranco Ciardo and Philippe Darondeau. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 444-454. ISBN: 978-3-540-31559-9.

[64] ProM. *ProM Tools*. https://promtools.org. En línea, último acceso: 24/04/2025.

[65] C.W. Günther and A. Rozinat. "Disco: discover your processes". En: *BPM (Demos)*. CEUR Workshop Proceedings. CEUR-WS, 2012.

[66] Fluxicon. *Disco*. https://fluxicon.com/disco/. En línea, último acceso: 24/04/2025.

[67] Alessandro Berti, Sebastiaan van Zelst and Wil Aalst. *Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science*. En línea, último acceso: 24/04/2025. Mayo de 2019. DOI: 10.48550/arXiv.1905.06169.

[68] Fraunhofer. *Pm4PY - State of the Art Process Mining in Python*. https://pm4py.fit.fraunhofer.de/. En línea, último acceso: 24/04/2025.

[69] Andrea Burattin. “Streaming Process Mining with Beamline”. English. En: *Proceedings of the ICPM Doctoral Consortium and Demo Track 2022*. CEUR Workshop Proceedings. ICPM 2022 Demo Track ; Conference date: 23-10-2022 Through 28-10-2022. CEUR-WS, 2022. URL: https://icpmconference.org/2022/.

[70] Sander J.J. Leemans. *Inductive visual Miner Manual*. English. 2017. 16 págs.

[71] Leemans. *Visual Miner*. https://leemans.ch/visualminer/home/. En línea, último acceso: 24/04/2025.

[72] Arya Adriansyah. “Aligning observed and modeled behavior”. En: 2014. URL: https://api.semanticscholar.org/CorpusID:64418472.

[73] Sander Leemans, Dirk Fahland and Wil Aalst. “Process and Deviation Exploration with Inductive Visual Miner”. En: vol. 1295. Enero de 2014, p. 46.

[74] MITRE. *MITRE Cyber Analytics Repository*. https://car.mitre.org. En línea, último acceso: 24/04/2025.

[75] Roman Daszczyszak et al. *TTP-Based Hunting*. Inf. téc. MITRE, 2019.

[76] *Hydra*. https://www.kali.org/tools/hydra/. En línea, último acceso: 24/04/2025.

[77] Openwall. *John the Ripper password cracker*. https://www.openwall.com/john/. En línea, último acceso: 24/04/2025.

[78] Carlospolop. *LinPEAS - Linux Privilege Escalation Awesome Script*. https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS. En línea, último acceso: 24/04/2025.

[79] Enisa. *European Union Agency for Cybersecurity*. https://www.enisa.europa.eu. Online, last accessed: 13/01/2025.

[80] Linux manual page. *Cron*. https://man7.org/linux/man-pages/man8/cron.8.html. Online, last accessed: 13/01/2025.

[81] Richard Weiss, Michael E. Locasto and Jens Mache. "A Reflective Approach to Assessing Student Performance in Cybersecurity Exercises". En: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. SIGCSE '16. Memphis, Tennessee, USA: Association for Computing Machinery, 2016, pp. 597-602. ISBN: 9781450336857. DOI: 10.1145/2839509.2844646. URL: https://doi.org/10.1145/2839509.2844646.

[82] Richard Weiss et al. "Cybersecurity Education and Assessment in EDURange". En: *IEEE Security Privacy* 15 (enero de 2017), pp. 90-95. DOI: 10.1109/MSP.2017.54.

[83] Valdemar Švábenskýet al. "Evaluating Two Approaches to Assessing Student Progress in Cybersecurity Exercises". En: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*. SIGCSE 2022. New York, NY, USA: Association for Computing Machinery, 2022, pp. 787-793. ISBN: 978-1-4503-9070-5. DOI: 10.1145/3478431.3499414. URL: https://doi.org/10.1145/3478431.3499414.

[84] Valdemar Švábenský et al. "Student assessment in cybersecurity training automated by pattern mining and clustering". En: *Education and Information Technologies* 27 (marzo de 2022), pp. 1-32. DOI: 10.1007/s10639-022-10954-4.

[85] Rapid7. *Metasploit Penetration Testing Software*. https://www.metasploit.com/. En línea, último acceso: 24/04/2025.

[86] Jan Vykopal et al. "Lessons learned from complex hands-on defence exercises in a cyber range". En: *2017 IEEE Frontiers in Education Conference (FIE)*. 2017, pp. 1-8. DOI: 10.1109/FIE.2017.8190713.

[87] Magdalena Granasen and Dennis Granåsen. "Measuring team effectiveness in cyber-defense exercises: a cross-disciplinary case study". En: *Cognition, Technology Work* 18 (febrero de 2016). DOI: 10.1007/s10111-015-0350-2.

[88] Zhihong Tian et al. "A Real-Time Correlation of Host-Level Events in Cyber Range Service for Smart Campus". En: *IEEE Access* 6 (2018), pp. 35355-35364. DOI: 10.1109/ACCESS.2018.2846590.

[89] Volatility Foundation. *Volatility Framework*. https://volatilityfoundation.org/the-volatility-framework/. En línea, último acceso: 24/04/2025.

[90] Kaie Maennel, Rain Ottis and Olaf Maennel. "Improving and Measuring Learning Effectiveness at Cyber Defense Exercises". En: *Secure IT Systems*. Ed. por Helger Lipmaa, Aikaterini MitrokotsaRaimundas Matulevičius. Cham: Springer International Publishing, 2017, pp. 123-138. ISBN: 978-3-319-70290-2.

[91] Chiara Braghin et al. "Towards the Monitoring and Evaluation of Trainees' Activities in Cyber Ranges". En: *Model-driven Simulation and Training Environments for Cybersecurity*. Ed. por George HatzivasilisSotiris Ioannidis. Cham: Springer International Publishing, 2020, pp. 79-91. ISBN: 978-3-030-62433-0.

[92] Mauro Andreolini et al. "A Framework for the Evaluation of Trainee Performance in Cyber Range Exercises". En: *Mobile Networks and Applications* 25 (febrero de 2020). DOI: 10.1007/s11036-019-01442-0.

[93] Andrea Artioli et al. "Evaluating Trainees in Large Cyber Exercises". En: *ITASEC24: Italian Conference on Cybersecurity*. Italy, 2024.

[94] Martin Macak, Radek Oslejsek and Barbora Buhnova. "Applying Process Discovery to Cybersecurity Training: An Experience Report". En: *2022 IEEE European Symposium on Security and Privacy Workshops (EuroSPW)*. 2022, pp. 394-402. DOI: 10.1109/EuroSPW55150.2022.00047.

[95] Martin Macak, Radek Oslejsek and Barbora Buhnova. "Process Mining Analysis of Puzzle-Based Cybersecurity Training". En: ITiCSE '22. Dublin, Ireland: Association for Computing Machinery, 2022, pp. 449-455. ISBN: 9781450392013. DOI: 10.1145/3502718.3524819. URL: https://doi.org/10.1145/3502718.3524819.

[96] Jason Diakoumakos et al. "Cyber-Range Federation and Cyber-Security Games: A Gamification Scoring Model". En: *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. 2021, pp. 186-191. DOI: 10.1109/CSR51186.2021.9527972.

[97] Iason Diakoumakos et al. “Cyber-security gamification in federation of cyber ranges: design, implementation, and evaluation”. En: *International Journal of Information Security* 24 (enero de 2025). DOI: 10.1007/s10207-024-00974-1.

[98] Amalia Damianou et al. “Situational Awareness Scoring System in Cyber Range Platforms”. En: septiembre de 2024, pp. 520-525. DOI: 10.1109/CSR61664.2024.10679451.

[99] Nikolaos Kakouros. “A cheat detection system for an educational pentesting cyber range: an intrusion deficit approach”. Available at https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-284254. Master thesis. KTH Royal Institute of Technology School of Electrical Engineering y Computer Science, 2020.

[100] Andrea Burattin. “Streaming Process Mining”. En: *Process Mining Handbook*. Ed. por Wil M. P. van der AalstJosep Carmona. Cham: Springer International Publishing, 2022, pp. 349-372. ISBN: 978-3-031-08848-3. DOI: 10.1007/978-3-031-08848-3_11. URL: https://doi.org/10.1007/978-3-031-08848-3_11.

[101] Rui Zhuang and Scott A. DeLoach and Xinming Ou. “Towards a Theory of Moving Target Defense”. En: *Proceedings of the First ACM Workshop on Moving Target Defense*. MTD ’14. Scottsdale, Arizona, USA: Association for Computing Machinery, 2014, pp. 31-40. ISBN: 9781450331500. DOI: 10.1145/2663474.2663479. URL: https://doi.org/10.1145/2663474.2663479.

[102] Sander J.J. Leemans. *Inductive visual Miner & Directly Follows visual Miner*. English. 2019. 19 págs.

# ANEXOS

# Anexo 1

# *Representación gráfica de modelos de procesos*

En este anexo se presenta la notación utilizada para la representación gráfica de los modelos de procesos construidos a partir de la aplicación del *plugin* Inductive Visual Miner de la herramienta ProM Tools. La información expuesta e imágenes utilizadas fueron extraídas de [102].

Un ejemplo de modelo obtenido por la herramienta se encuentra representado en la Figura A.1. Si bien estos modelos están basados en árboles de procesos, su representación como tal no es un árbol de procesos. Los árboles de proceso son estructuras jerárquicas construidas en base a nodos conformados por hijos. Los nodos expresan cierto significado en base a los hijos, y dicho significado dependerá del operador utilizado. Existen seis tipos de operadores:

- *xor*: solamente uno de los hijos debe ser ejecutado.
- *sequence*: todos los hijos deben ser ejecutados siguiendo el orden específico.
- *interleaved*: todos los hijos deben ser ejecutados pero no pueden superponerse en el tiempo y no define un orden en particular.
- *concurrent*: todos los hijos deben ser ejecutados, sin importar si se superponen en el tiempo y el orden de ejecución.
- *or*: al menos uno de los hijos debe ser ejecutado. Podría darse el caso en que todos los hijos sean ejecutados, en dicho caso no importa el tiempo ni orden de ejecución de los hijos.
- *loop*: el primer hijo debe ser ejecutado. Luego de dicha ejecución hay una posible elección entre terminar la ejecución o ejecutar el segundo

hijo y luego el primer hijo. En este punto se presenta la misma decisión nuevamente.

En el caso de la Figura A.1, el árbol de proceso asociado a dicha representación es: $sequence(xor(a, b), concurrent(c, d), interleaved(e, f))$.

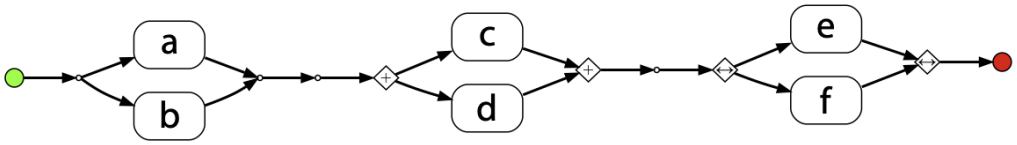



**Figura A.1:** Ejemplo modelo de proceso. Extraído de [102].

Los distintos constructores se observan en la Figura A.2. A continuación se brinda más detalle de cada uno de estos.

- *Source*: es el punto de origen.
- *Sink*: es el punto de finalización.
- *Exclusive choice*: misma semántica que el operador *xor*.
- *Concurrency*: misma semántica que el operador *concurrent*.
- *Interleaving*: misma semántica que el operador *interleaved*.
- *Inclusive choice*: misma semántica que el operador *or*.

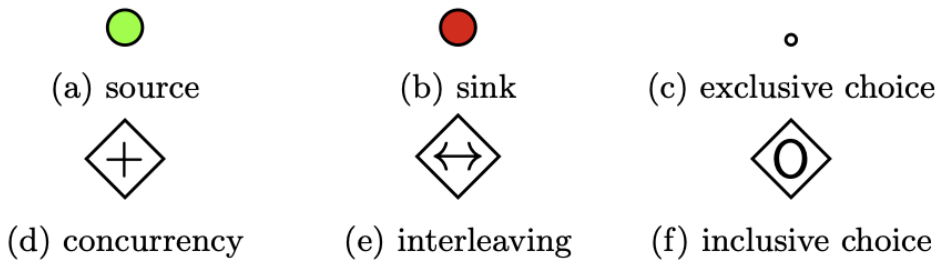



**Figura A.2:** Notación utilizada para los modelos. Extraído de [102].

En definitiva, en el modelo de ejemplo se observa que primero se debe ejecutar A o B. Luego, deben ser ejecutados tanto C como D sin importar el orden y tiempo de ejecución, y por último, tanto E como F deben ser ejecutados sin importar su orden pero cumpliendo que no se superpongan en el tiempo.