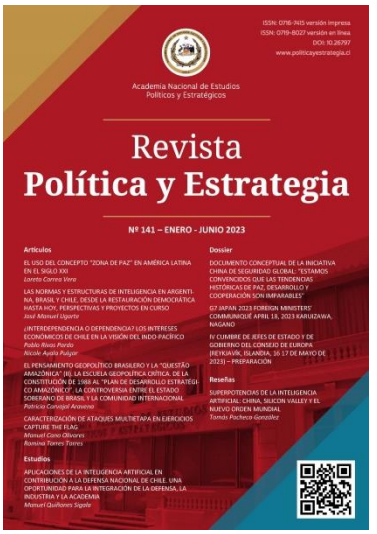


Revista Política y Estrategia
Nº 141, (2023)

Editada por: **Academia Nacional de Estudios Políticos y Estratégicos (ANEPE) Chile**.

Lugar de edición: Santiago, Chile

Dirección web:
http://www.politicayestrategia.cl

ISSN versión digital: 0719-8027

ISSN versión impresa: 0716-7415

DOI: **https://doi.org/10.26797/rpye.vi141.1028**

Para citar este artículo / To cite this article: Cano Olivares, Manuel y Torres Torres, Romina: "Caracterización de ataques multietapa en ejercicios Capture The Flag".

Revista Política y Estrategia N° 141. 2023. pp. 133-151

DOI: **https://doi.org/10.26797/rpye.vi141.1028**

Si desea publicar en Política y Estrategia, puede consultar en este enlace las Normas para los autores:

To publish in the journal go to this link:

http://politicayestrategia.cl/index.php/rpye/about/submissions#authorGuidelines

CARACTERIZACIÓN DE ATAQUES MULTIETAPA EN EJERCICIOS CAPTURE THE FLAG ∞

MANUEL CANO OLIVARES •
ROMINA TORRES TORRES • •

RESUMEN

Los ciberataques sufridos por las organizaciones son por naturaleza ataques multietapa, también conocidos como MSNAs, se componen de una serie de pasos correlacionados en el tiempo para lograr un objetivo específico. Comprender y analizar estos ataques plantea desafíos significativos en la detección y defensa efectiva. Sin embargo, la escasez de ejemplos reales de MSNAs disponibles para la investigación y análisis complica el estudio de estos ataques. En este artículo, se propone una metodología novedosa para caracterizar los MSNAs utilizando un modelo simplificado de Cyber Kill Chain y archivos históricos de eventos de captura de bandera (CTF) liberados por DEF CON. Proponemos un método que aplicamos a los archivos históricos de DEF CON 22 con la que logramos caracterizar visualmente 148 MSNAs dirigidos al equipo ganador. Los resultados revelaron una secuencia clara de etapas en los ataques, proporcionando una comprensión más profunda.

Palabras clave: *Ciberseguridad; CTF; Cyber Kill Chain; ataque de red de múltiples etapas; reconstrucción de ataques.*

CHARACTERIZATION OF MULTISTAGE ATTACKS IN CAPTURE THE FLAG EXERCISES

• Magíster en Ingeniería Informática, Universidad Andrés Bello; Ingeniero de Ejecución en Informática, Pontificia Universidad Católica de Valparaíso; Diplomado en Gestión de Tecnologías de Información, Universidad Andrés Bello; Diplomado en Ciberseguridad, Academia Nacional de Estudios Políticos y Estratégicos (ANEPE), actualmente se desempeña en la Facultad de Ingeniería de la Universidad Andrés Bello, Santiago, Chile. m.canoolivares@uandresbello.edu ORCID: https://orcid.org/0000-0002-7536-6715

• • Doctora en Ingeniería Informática, Universidad Técnica Federico Santa María. Desde los años 2018 y 2023 (marzo) se ha desempeñó como directora del Magíster en Ciencias de la Computación y del Magíster en Gestión de TI y Telecomunicaciones - Nacional - Facultad de Ingeniería y Director de Magíster en Gestión de TI y Telecomunicaciones – Sede Viña del Mar – Facultad de Ingeniería. Universidad Andrés. Bello. Actualmente cumple funciones como académica en la Facultad de Ingeniería y Ciencias de la Universidad Adolfo Ibánez. Chile. romina.torres.t@uai.cl . ORCID: https://orcid.org/0000-0003-2705-4298

∞ Fecha de recepción: 030623 - Fecha de aceptación: 300623.

ABSTRACT

The cyberattacks suffered by organizations are inherently multistage attacks, also known as MSNAs (Multistage Network Attacks). They consist of a series of correlated steps over time to achieve a specific objective. Understanding and analyzing these attacks pose significant challenges in detection and effective defense. However, the scarcity of real MSNA examples available for research and analysis complicates the study of these attacks. In this article, a novel methodology is proposed to characterize MSNAs using a simplified model of the Cyber Kill Chain and historical Capture the Flag (CTF) event files released by DEF CON. We propose a method that we applied to the historical files of DEF CON 22, through which we successfully visually characterized 148 MSNAs targeting the winning team. The results revealed a clear sequence of stages in the attacks, providing a deeper understanding.

Key words: *Cybersecurity; CTF; Cyber Kill Chain; multi-stage attack; attack reconstruction.*

CARACTERIZAÇÃO DE ATAQUES MULTIESTÁGIOS EM EXERCÍCIOS CAPTURE THE FLAG

RESUMO

Os ataques cibernéticos sofridos pelas organizações são por natureza ataques de vários estágios, também conhecidos como MSNAs, são compostos por uma série de etapas correlacionadas no tempo para atingir um objetivo específico. Compreender e analisar esses ataques apresenta desafios significativos na detecção e defesa eficazes. No entanto, a escassez de exemplos reais de MSNAs disponíveis para pesquisa e análise complica o estudo desses ataques. Neste artigo, uma nova metodologia é proposta para caracterizar MSNAs usando um modelo simplificado de Cyber Kill Chain e arquivos históricos de eventos capture the flag (CTF) divulgados pelo DEF CON. Propomos um método que aplicamos aos arquivos históricos do DEF CON 22 com o qual conseguimos caracterizar visualmente 148 MSNAs direcionados ao time vencedor. Os resultados revelaram uma sequência clara de etapas nos ataques, proporcionando uma compreensão mais profunda.

Palavras-chave: *Cibersegurança; CTF, Cyber Kill Chain; ataque de rede em vários estágios; reconstrução de ataque.*

I. INTRODUCCIÓN

Un ciberataque se define como un "intento de destruir, exponer, alterar, inhabilitar, robar u obtener acceso no autorizado o hacer un uso no autorizado de un activo"[1]. Con el aumento de los ataques, el estudio de los ciberataques ha adquirido gran relevancia a nivel social. Los ataques de red son inherentemente multietapa o multipaso (MSNA)[2], lo que significa que constan de al menos dos ataques distintos relacionados. Estos ataques pueden durar horas, días o meses hasta alcanzar su objetivo. Las organizaciones más complejas que son víctimas de un MSNA suelen contar con Centros de Operación de Seguridad (SOCs). Estos SOCs están dedicados a detectar y mitigar los daños causados por estos ataques en una compleja topología de red vigilada. Los SOC utilizan modelos o marcos de trabajo para analizar y categorizar las tácticas, técnicas y procedimientos utilizados en los ciberataques. El Cyber Kill Chain (CKC) desarrollado por Hutchins, Cloppert y Amín[3], ilustrado en la Figura 1, está basado en la táctica Kill Chain del F2T2EA utilizado por el ejército de los Estados Unidos. Es un marco de trabajo ampliamente utilizado para este tipo de ataques. El CKC está orientado a desactivar amenazas del tipo "Amenaza Avanzada Persistente" (APT) y recopila inteligencia sobre las amenazas y su uso futuro, permitiendo que el atacante continúe sus actividades después de ser detectado.

Figura 1
Cyber Kill Chain (CKC)

Existen variaciones del CKC, como el modelo Unified Kill Chain (UKC) propuesto por Pols[4], que propone la evolución de un ataque usando 18 etapas. También, Singh, Callupe y Govindarasu[5] propusieron un modelo simplificado del CKC compuesto por cuatro etapas: reconocimiento, acceso, ejecución y persistencia, de acuerdo con la Figura 2.

1 International Organization for Standardization [ISO]. ISO/IEC. 27000:2016. Information technology — Security techniques — Information security management systems — Overview and vocabulary [en línea]. 2016. Disponible en: https://www.iso.org/standard/66435.html [Consulta: 29 de agosto de 2021].

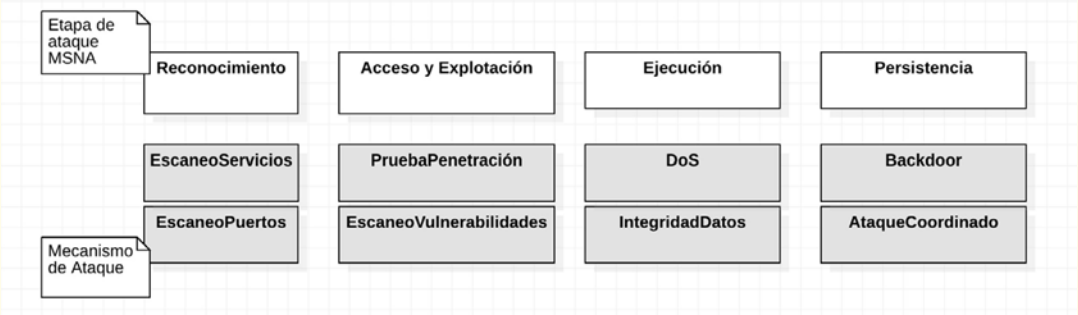
2 NAVARRO, Julio, DERUYVER, Aline y PARREND, Pierre. 2018. A systematic survey on multi-step attack detection. Computers Security [en línea], vol. 76, 214-249. Disponible en: https://doi.org/10.1016/j.cose.2018.03.001 [Consulta: 29 de agosto de 2021].

3 HUTCHINS, Eric, CLOPPERT, Michael y AMIN, Rohan. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research [en línea], vol. 1, no. 1, 1-14. Disponible en: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf [Consulta: 29 de agosto de 2021].

4 POLS, Paul. 2022. The Unified Kill Chain [en línea]. Disponible en: https://www.unifiedkillchain.com/assets/The-Unified-Kill-Chain.pdf [Consulta: 24 de mayo de 2023].

5 SINGH, Vivek, CALLUPE, Steven y GOVINDARASU, Manimaran. Testbed-based Evaluation of SIEM Tool for Cyber Kill Chain Model in Power Grid SCADA System. En: 2019 North American Power Symposium (NAPS) [en línea]. Kansas, Estados Unidos, octubre de 2019, pp. 1-6. Disponible en: https://www.researchgate.net/publication/336623025_Testbed-based_Evaluation_of_SIEM_Tool_for_Cyber_Kill_Chain_Model_in_Power_Grid_SCADA_System [Consulta: 29 de agosto de 2021].

Figura 2.
CKC simplificado con procesos y mecanismos de ataque



La definición de cada uno de los pasos del CKC simplificado de Singh y sus coautores[6] es la siguiente:

Reconocimiento: el atacante trata de recopilar información relevante sobre la red y sus servicios para identificar posibles víctimas y objetivos de ataque. Para ello, puede usar herramientas de red como ping, arp, traceroute, nmap, entre otras.

Acceso: el atacante intenta conectarse a los objetivos seleccionados para descubrir sus posibles vulnerabilidades. Esta información se usará más adelante para lograr acceso y/o escalada de privilegios. En esta etapa, pueden utilizarse herramientas como OpenVAS, Metasploit, Nessus, entre otras.

Lanzamiento/ejecución del ataque: en esta etapa, el atacante intenta explotar las vulnerabilidades encontradas en la red o sus componentes para obtener control sobre ellos. Las actividades pueden incluir virus, gusanos, caballos de Troya, ataques de denegación de servicio (DoS), ataques de Man-in-the-Middle (MITM), violaciones a la integridad y violaciones a la privacidad.

Persistencia: en esta etapa, el atacante crea una puerta trasera para mantener su acceso persistente y poder ingresar en el futuro para repetir el ataque o lanzar múltiples ataques en diferentes plataformas de manera coordinada. La persistencia requiere que el lanzamiento/ejecución se haya realizado con éxito.

Los ejercicios de ciberseguridad tipo Capture the Flag (CTF) han tomado gran relevancia en la demostración de capacidades de detección y respuesta por parte de los SOCs. Según ŠVÁBENSKÝ, ČELEDA, VYKOPAL y BRIŠÁKOVÁ[7], los participantes pueden aprender habilidades técnicas como criptografía y seguridad de redes, así como aspectos humanos como la ingeniería social y la conciencia de seguridad cibernética. Dado esto, han sido utilizados exitosamente para entrenar profesionales en seguridad de una manera lúdica, tanto

6 Ibíd.

7 ŠVÁBENSKÝ, V., ČELEDA, P., VYKOPAL, J. y BRIŠÁKOVÁ, S., 2021. Cybersecurity knowledge and skills taught in capture the flag challenges. Computers & security [en línea], vol. 102, no. 102154, ISSN 0167-4048. DOI 10.1016/j.cose.2020.102154. Disponible en: https://www.sciencedirect.com/science/article/pii/S0167404820304272 [Consulta: 24 de mayo de 2023]

por instituciones educacionales[8] como por gobiernos y compañías privadas. Un ejemplo de esto último son los organizados por compañías como Google en 2020 (https://capturetheflag.withgoogle.com/) y Facebook en 2019 (https://www.facebook.com/notes/facebook-bug-bounty/announcing-facebook-ctf-2019/2629218463759030/). Existen diferentes tipos de CTFs[9]. Los más conocidos son los del tipo jeopardy (donde es un entorno controlado del equipo contra la máquina) y los de tipo ataque/defensa, en los que nos concentramos en este trabajo. En CTFs ataque/defensa, cada equipo compuesto de hasta 5 miembros tiene una máquina y/o subred, la cual posee tanto programas con vulnerabilidades que deben explotarse para obtener las banderas ocultas. Cada equipo tiene un tiempo corto para parchar sus servicios y desarrollar exploits para luego conectarse con los otros equipos participantes intentando capturar sus banderas. No está permitido bloquear programas vulnerables lo cual es monitoreado durante la competencia. Esta competencia tiene puntos. Cada equipo anota un punto cada vez que logra defender su máquina o atacar la máquina de otros equipos. La defensa otorga puntos, cada equipo tiene la posibilidad de defender su sistema, cuando el equipo está buscando un *exploit* o vulnerabilidad se puede incluir un parche en el servicio para proteger cualquier fuga de información que pueda ocurrir. Un CTF tiene rondas (por ejemplo de 5 minutos), por lo que en cada una es posible seguir obteniendo puntos si las vulnerabilidades no han sido parchadas en anteriores. El ganador de la competencia es el equipo con la mayor cantidad de puntos.

El DEF CON CTF es conocido por ser uno de los CTF más largos y desafiantes en la comunidad de seguridad informática. En general, participan los 20 mejores equipos del mundo compuesto de 5 miembros. Este CTF es del tipo Ataque-Defensa, es altamente competitivo y riguroso. El primer CTF se realizó en el DEF CON 4 (1996). Su formato actual comenzó en DEF CON 10 (2002). A partir de entonces, se refinó aún más en DEF CON 13 (2005) cuando se enfatizó en la explotación binaria y su corrección. Finalmente, desde DEF CON 25 (2017), se utiliza el emulador cLEMENCy (LEgitbs Middle ENdian Computer) desde el cual se ejecutan los servicios vulnerables para que los participantes estén obligados a prescindir de herramientas avanzadas de corrección de *software* que podrían favorecer a algún equipo en particular. Dado que en este CTF para que los equipos logren obtener las banderas deben realizar ataques por naturaleza MSNA, la estrategia de este trabajo es caracterizar los MSNAs que se dan en estos eventos. Para ello, hemos primero analizado la literatura respecto de la reconstrucción de ataques multietapa, segundo hemos estudiado cómo obtener de la data de estos CTFs datos en un formato procesable y como tercer paso nos hemos centrado en la caracterización de un evento específico.

Respecto de la literatura destacamos un enfoque basado en meta-alertas correlacionadas, así como alertas únicas no agrupadas para construir grafos de escenarios de APT[10].

8 MIRKOVIC, J. y PETERSON, P. 2014. Class Capture-the-Flag Exercises. En: 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14) [en línea]. San Diego, California, agosto de 2014. Disponible en: https://www.usenix.org/biblio/class-capture-flag-exercises [Consulta: 29 de agosto de 2021].

9 KUCEK, S. y LEITNER, M., 2020. An empirical survey of functions and configurations of open-source capture the flag (CTF) environments. Journal of network and computer applications [en línea], vol. 151, no. 102470, ISSN 1084-8045. DOI 10.1016/j.jnca.2019.102470. Disponible en: http://dx.doi.org/10.1016/j.jnca.2019.102470.

10 WILKENS, Florian "et al". Multi-Stage Attack Detection via Kill Chain State Machines [en línea]. 2021. Disponible en: https://arxiv.org/abs/2103.14628 [Consulta: 29 de agosto de 2021].

Esto se logra construyendo una Máquina de Estados de Kill Chain (KCSM) que opera con datos de alerta agrupados para identificar estados y transiciones de ataques de múltiples etapas. Los grafos de escenarios de APT resultantes de este proceso visualizan posibles campañas de APT en la red y proporcionan un contexto procesable durante las investigaciones. Otro enfoque ha sido utilizar reconocimiento de escenarios de ataque basados en pasos de ataque, lo que permite ofrecer un enfoque flexible para reconocer ataques[11]. El sistema que proponen puede aportar un valor añadido en investigaciones forenses y en *honeypots* de investigación. Su arquitectura usa seis componentes principales: receptor de alertas, normalización de alertas, preprocesamiento de alertas, agrupación de alertas, reducción de alertas y reconocimiento de escenarios de ataque. Una estrategia recurrente ha sido utilizar un sistema de detección de intrusos (IDS) a partir del registro de tráfico de red de estos eventos para la generación de alertas como paso inicial para la reconstrucción y caracterización[12].

Este trabajo se divide en las siguientes secciones. Sección II: "Método propuesto para la caracterización de un CTF" donde utilizamos como base el IDS, Snort, para generar un conjunto de alertas a partir de archivos pcaps liberados por la conferencia, luego utilizamos un modelo CKC para clasificar cada alerta y luego una componente visual para presentar los MSNAs; Sección III: "Resultados", donde se muestran los resultados aplicando este método al CTF 22 de la DEF CON; y Sección IV, donde se establecen las conclusiones y el potencial trabajo futuro.

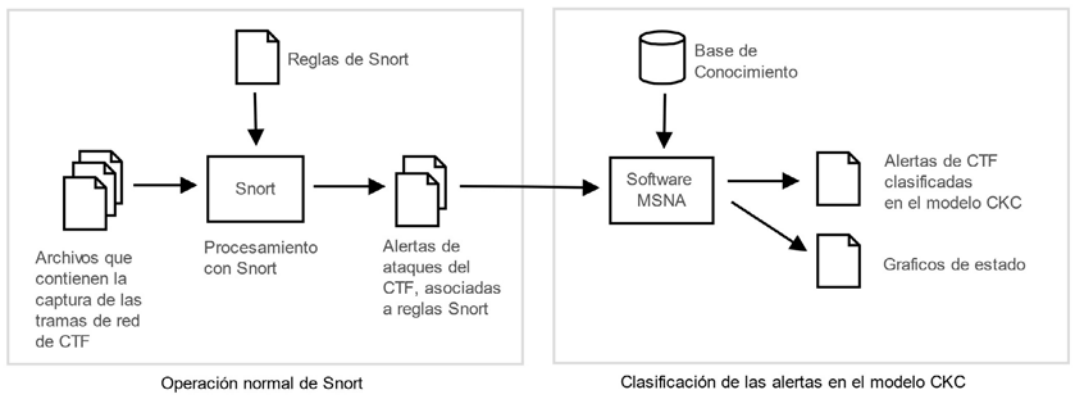
II. MÉTODO PROPUESTO PARA LA CARACTERIZACIÓN DE UN CTF

En esta investigación se propone utilizar el modelo Kill Chain Simplificado en Máquinas de Estado (KCS2ME), es decir, emplear máquinas de estado para representar los pasos del modelo de CKC simplificado visto anteriormente.

11 MENDES, J. y SOARES, R. 2019. Flexible Approach to Multi-Stage Network Attack Recognition. International Journal of Computer Science and Information Security (IJCSIS) [en línea], vol. 17, no. 8, 67-73. Disponible en: https://www.academia.edu/40458556/Flexible_Approach_to_Multi_Stage_Network_Attack_Recognition [Consulta: 29 de agosto de 2021].

12 JULISCH, Klaus. 2003. Clustering intrusión detection alarms to support root cause analysis. ACM Transactions on Information and System Security [en línea], vol. 6, no. 4, 443-471. Disponible en: https://doi.org/10.1145/950191.950192 [Consulta: 29 de agosto de 2021].

Figura 3
Método general de caracterización de ataques multietapa en ejercicios CTF



Fuente: elaboración propia.

La Figura 3 muestra la propuesta de método para caracterizar ataques multietapa en ejercicios CTF utilizando un enfoque forense basado en alertas en vez de las tramas directas de la red. Esta propuesta consta de dos bloques principales, el primero es la operación de Snort de manera normal sobre las capturas de red y el segundo es la clasificación de las alertas obtenidas según el modelo CKC. La primera parte, que implica la operación normal de Snort, requiere simplemente la selección del CTF a analizar y genera alertas a partir de las capturas de las tramas de red registradas durante el CTF. La segunda requiere el apoyo del *software* (https://github.com/communitylab4u/MSNA/) y la clasificación previa de las reglas de Snort a utilizar mediante el modelo CKC, que se encuentran disponibles con el *software*. Luego, se ejecuta la clasificación de las alertas y la caracterización visual de los CTF.

Por lo tanto, en el bloque de la izquierda se muestra el primer componente del método: Snort, el cual es de fuente abierta, no requiere un pago por licencia y viene integrado con Linux Ubuntu. La entrada al componente Snort incluye las reglas Snort actualizadas previamente en su sitio web y los archivos que contienen las capturas de las tramas de red de CTF, que generalmente están disponibles en formato PCAP (por ejemplo: https://defcon.org/html/links/dc-ctf.html). Para procesar el *dataset*, se ejecuta el siguiente comando: snort -c /etc/snort/snort.conf -r archivo.pcap

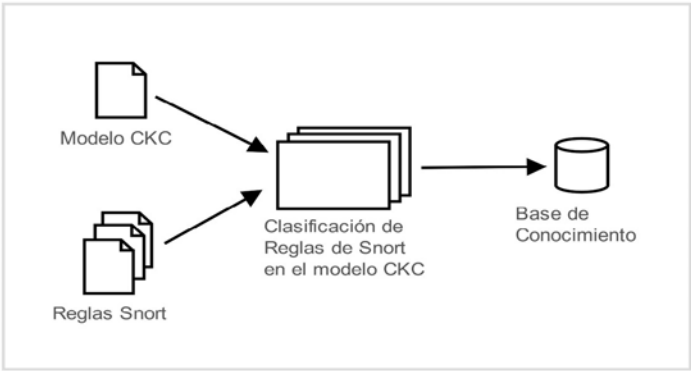
Por otro lado, en el bloque derecho de la Figura 3 se pueden observar dos componentes adicionales: Base de Conocimiento y *Software* MSNA (repositorio de clasificación de alertas y conjunto de scripts que se encuentran disponibles en https://github.com/communitylab4u/MSNA/). Para el componente Base de Conocimiento, dado que las reglas de Snort son generales, una parte del método consiste en clasificar las reglas en las etapas del modelo CKC, tal como se muestra en la Figura 4 y se describe en el Algoritmo 1. Para ello, primero se genera una plantilla vacía utilizando un *script* que toma las reglas instaladas dentro de Snort, a partir del cual se extrae el SID de cada regla y se generan los campos necesarios para el siguiente paso.

Algoritmo 1. Algoritmo utilizado para la clasificación CKC de las alertas de los *dataset*

Algoritmo 1	
Actividad 1	Recorre el archivo de alertas, una por una.
Actividad 1.1	Por cada alerta se obtiene su clasificación de la etapa CKC a la que pertenece basada en el indicador que se almacena en el archivo, o mediante reglas: Regla 1: una conexión exitosa tras un ataque de desbordamiento de buffer, o un exploit, se considerará como sospechosa de ser parte de un ataque que se encuentra en la etapa 4 del CKC. Regla 2: una serie de ataques de etapa 3 pueden detenerse debido a que el atacante considera que no se puede vulnerar o bien porque ya logró hacerlo, esto último es especialmente cierto si hubo ataques de paso 2 previamente en los que el atacante obtuvo las versiones de cada servicio y tiene un buen supuesto acerca de que ataque puede funcionar y que no, de allí que se considerará como sospechosa de ser parte de un ataque que se encuentra en la etapa 4 del CKC.
Actividad 1.2	Por cada alerta clasificada se actualizan el repositorio de alertas clasificadas y los contadores de resumen por host y atacante.
Actividad 2	Se guardan en CSV el repositorio de alertas clasificadas y los contadores de resumen por host y atacante.
Actividad 3	Se generan gráficos que permiten una mejor comprensión de lo ocurrido.

Fuente: elaboración propia.

Figura 4.
Generación de la Base de Conocimiento usando los métodos de clasificación



Fuente: elaboración propia.

El componente Base de Conocimiento se refiere al repositorio que almacena tuplas de regla-etapa. Las técnicas utilizadas para realizar la clasificación de las reglas Snort, dentro de una etapa del CKC, son las siguientes:

Basado en el código de la vulnerabilidad: muchas reglas incluyen en su descripción un código de bases de datos de vulnerabilidades como CVE, bugtraq, msb, arachNIDS, osvdb, McAfee, Nessus o una URL que permite obtener información acerca de lo que la regla busca detectar y así clasificarla en el CKC. Un ejemplo de esto se puede ver en la regla 100000223, cuya descripción es *reference:bugtraq,16213*, lo que permite encontrar información en https://web.archive.org/web/20160423010632/https://www.securityfocus.com/bid/16213/

Basado en la definición de la regla: la regla Snort contiene las condiciones en las que debe alertar algún tipo de ataque, y al interpretar estas condiciones se puede hacer la correspondiente clasificación en el modelo CKC, por ejemplo:

alert tcp $EXTERNAL_NET any - $HTTP_SERVERS $HTTP_PORTS (msg:“WEBATTACKS nmap command attempt”, flow: to_server, established, content: “nmap%20”, nocase, classtype: web application attack, sid:1361, rev:5,)

Allí se indica que se debe generar una alerta cuando un cliente se identifica a sí mismo como “nmap” al conectarse desde el exterior hacia el servidor web, lo que puede interpretarse como un intento de escanear servicios web para obtener información sobre

la versión del servidor. Esto se considera como una etapa 2 del modelo CKC.

Basado en la descripción: en la descripción de la regla puede haber un texto que indica de manera certera su naturaleza. Los más importantes son: XSS, Overflow, SQL injection, exploit.

Basado en el uso de herramientas: esto corresponde al uso de herramientas que se sabe qué tipo de resultados generan. Por ejemplo, haciendo reconocimiento de red, usando ping o traceroute, generará alertas que se pueden clasificar como de descubrimiento de red, es decir, de etapa 1 del CKC usado.

Basado en el nombre del archivo: las reglas de Snort se agrupan en archivos por temática, lo que es útil en algunos casos, mientras que en otros no. Entre los nombres de archivo que pueden ayudar a clasificar se encuentran los siguientes: Virus, Backdoor y Shellcode. Estos implican compromiso del *host*, es decir, etapa 4 del CKC. En cambio, otros nombres como Web o Misc no permiten hacer una adecuada clasificación.

Una vez que el *software* MSNA procesa las tramas de red de un CTF, se generará un conjunto de alertas clasificadas en el modelo CKC. La salida incluirá un archivo con los Snort ID (SID) de las reglas que no estaban clasificadas en el CKC, además de la cantidad de veces que se encontraron en el archivo de alertas.

Para ello, se realiza el supuesto de que el destino del ataque es un servidor o un grupo de servidores en producción y, por tanto, no tienen usuarios haciendo uso de este en su consola, como en el caso de un equipo de escritorio. Esto implica lo siguiente: 1) Las conexiones salientes son automatizadas, por lo que normalmente una conexión saliente será exitosa y las credenciales de usuario y contraseña serán correctas. Otros comportamientos, como un barrido al exterior, se consideran bajo el control de un atacante; 2) No hay sistemas de chat en uso. Por esta razón, una conexión de chat de salida se considerará una acción de comando y control; 3) Una conexión saliente de un virus implica que el servidor está comprometido; y 4) Una conexión desde el exterior a un servicio del servidor que debería entregarse solo a equipos de la misma LAN se considerará un ataque. Ejemplos de estos servicios son compartir carpetas e impresoras usando protocolo SMB, así como el servicio de proxy. En este último caso, al estar el cliente fuera de la LAN, se considera que esta conexión es una búsqueda de proxy reverso para aprovecharlos de alguna manera (por ejemplo, un ataque de falsa bandera o un salto hacia el interior de la organización).

III. RESULTADOS

DEF CON 22 CTF

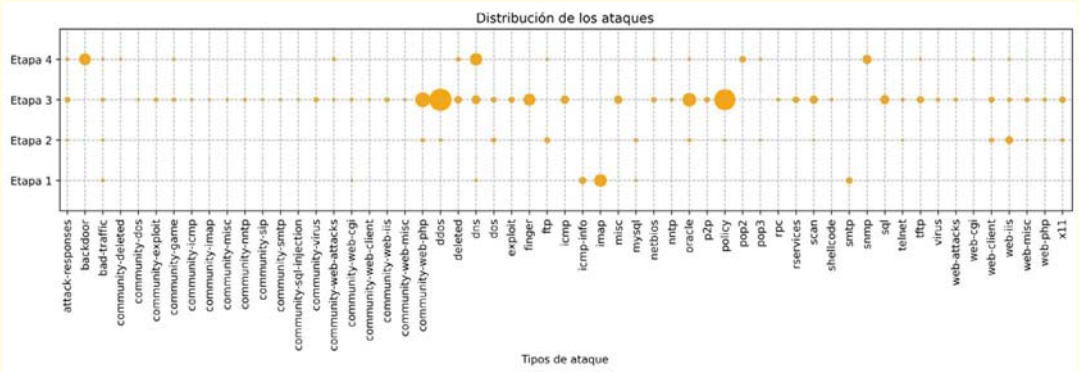
En esta investigación hemos decidido caracterizar la DEF CON 22 CTF (2014) porque es la última versión del evento en la que los ataques se detectaron a través de las reglas de Snort. DEF CON 22 CTF se llevó a cabo del 8 al 10 de agosto de 2014. El CTF se realizó por tres días, entre las 9:00 y las 20:00 horas, iniciando la conexión entre distintas subredes a las 10:00 horas en horario de verano del Pacífico en USA (PDT), UTC -7. A cada uno de los 20 equipos participantes se le asignó una red clase C y un host para defender, y los miembros de cada equipo se repartieron las tareas de ataque y defensa durante la competencia. Cada

host tenía cuatro servicios vulnerables[13]: *eliza*, un simulador económico espacial basado en texto, *wdub*, un servicio web, *justify*, un solucionador de restricciones e *IMAP*, un servidor de correo electrónico. Además de defender su propio host y atacar a otros equipos con la misma configuración, también hubo un desafío de hardware. Las banderas se obtenían al explotar vulnerabilidades en los servidores. Los equipos debían defender sus propias banderas y obtener las de sus oponentes, desarrollando parches binarios y llevando a cabo ataques. Según Yam[14] las estrategias utilizadas en los CTF son: Reutilización de exploits, Detección de exploits, Detección de banderas salientes, Hazañas ofuscadas, Ofuscación del tráfico de exfiltración, Destrozar banderas (modificarlas), Colusión y Apalancamiento en recursos externos.

Preparación de Ambiente

Para poder lograr el objetivo de este trabajo, se levantó un laboratorio con Ubuntu 20.04.2.0 LTS, que correspondía a la última versión disponible del sistema operativo a la fecha. Debido a que los ataques aprovechaban vulnerabilidades presentes en bases de datos públicas, se crearon reglas en Snort para detectar dichos ataques. Por tanto, esta versión incluye Snort 2.9.7.0 y se utilizaron las reglas snortrules-snapshot-29171.tar.gz, las cuales se descargaron posterior al registro en la web de Snort (https://www.snort.org/). En la Figura 5 se muestra un resumen de la clasificación por etapa CKC de las reglas que utiliza Snort para detectar ataques.

Figura 5.
Distribución de la clasificación por etapa CKC y tipo de reglas utilizadas por Snort para detectar ataques



Fuente: elaboración propia.

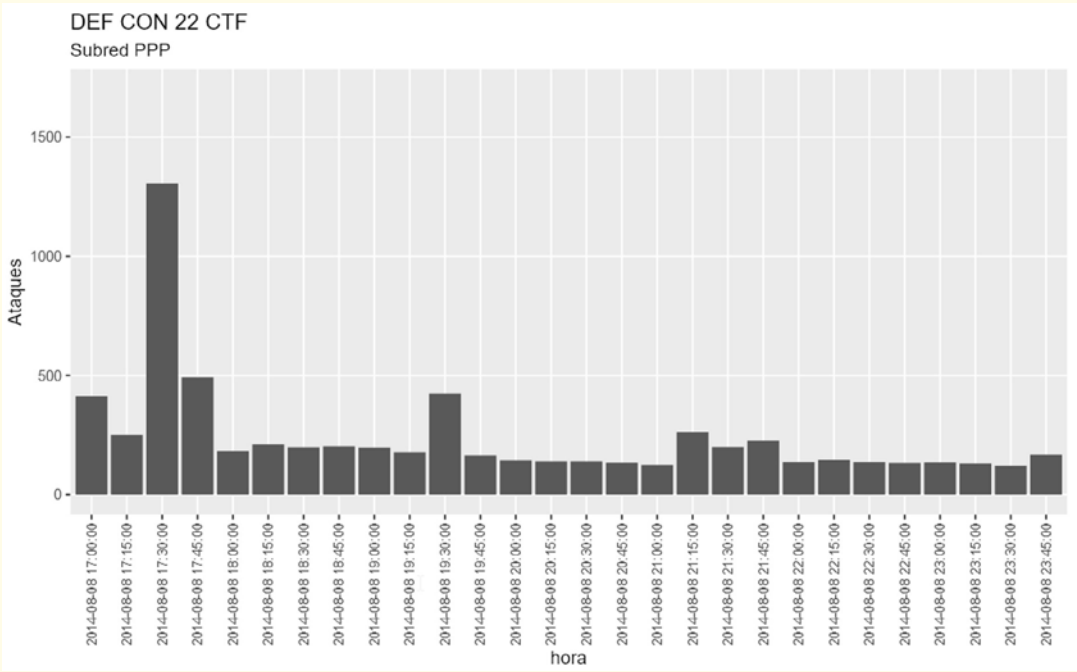
Las capturas de red de los CTFs pueden descargarse en formato CAP desde https://media.defcon.org/DEF%20CON%2022/DEF%20CON%2022%20ctf/. Sin pérdida de generalidad, solo se utilizaron las capturas de la subred del equipo ganador del evento, Plaid Par-

13 Stratum 0, 2014. Def con 22 capture the flag. Disponible en: https://stratum0.org/blog/posts/2014/08/29/defcon-ctf-2014/

14 YAM, W.K.J., 2016. *Strategies used in capture-the-flag events contributing to team performance.* S.l.: Monterey, California: Naval Postgraduate School. [en línea]. Disponible en: https://hdl.handle.net/10945/48498 [Consulta: 23 de mayo de 2023].

liament of Pwning (PPP), las cuales tienen un tamaño comprimido de 4.47 GB. Se descargó esta información utilizando el archivo torrent proporcionado por el sitio web, puesto que admite la recuperación de descargas interrumpidas. El *dataset* incluyen las conexiones con origen o destino desde 10.5.1.0/24. Algunas de las conexiones registradas en la subred corresponden a ataques al servidor 10.5.1.2, mientras que otras corresponden a ataques desde algún equipo de la subred hacia los otros servidores objetivos. También hay conexiones al servidor que mantiene el conteo del puntaje y verifica las banderas, llamado Scorebot, y navegación normal desde los equipos de la subred. Al momento de procesar las alertas, el servidor donde está Snort tenía configurado horario de invierno de Chile continental (UTC-4), por lo que las alertas se generaron con una diferencia de 3 horas. En la Figura 6 se observa la frecuencia de los ataques agrupada en bloques de 15 minutos, en donde los ataques en el bloque de las 17:30 horas sobresalen del resto.

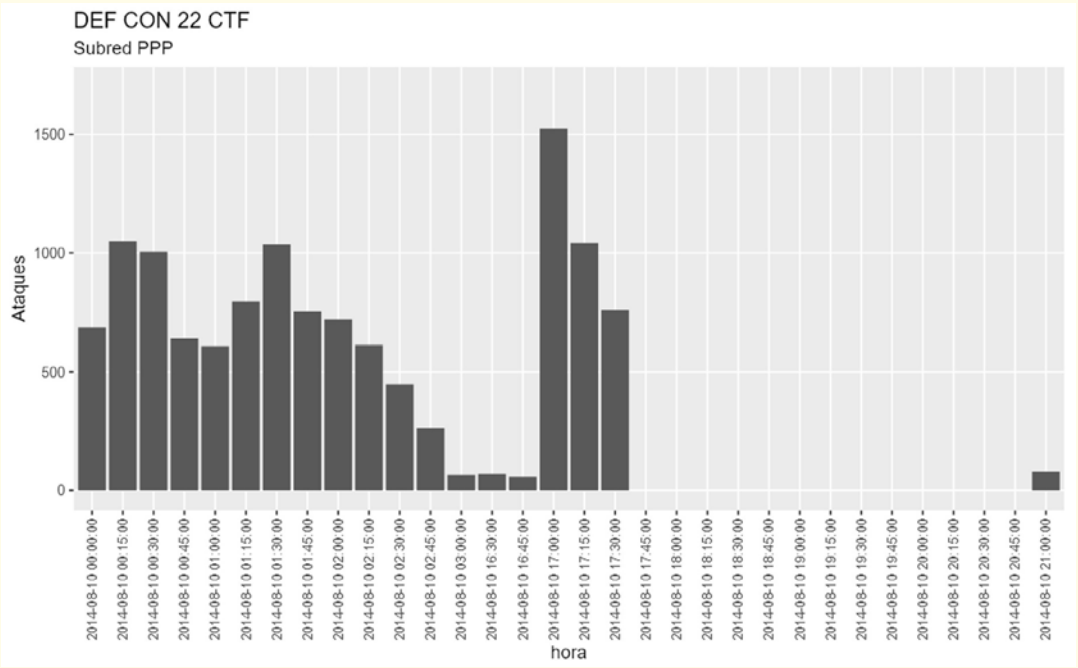
Figura 6.
Ataques registrados por Snort en la subred 10.5.1.0/24 el primer día de competencia



Fuente: elaboración propia. Generado utilizando las alertas obtenidas por SNORT.

Por otra parte, en la Figura 7 se puede apreciar un alza en los ataques poco antes del fin del evento.

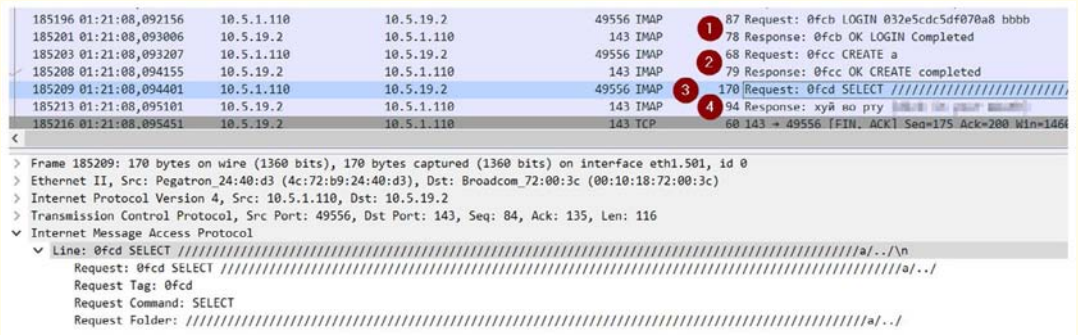
Figura 7.
Ataques registrados por Snort en la subred 10.5.1.0/24 el tercer y último día de competencia



Fuente: elaboración propia. Generado utilizando las alertas obtenidas por Snort.

Utilizando Wireshark para analizar el archivo de captura de red "ppp_00100_20140809181755.cap" y empleando el filtro "*(ip.addreq 10.5.1.110 and ip.addreq 10.5.19.2) and (tcp.port eq 143)*", se pueden ver cuatro acciones que conducen a un ataque exitoso, las cuales se muestran en la Figura 8: 1. Inicio de sesión, 2. Creación de un mailbox, 3. Seleccionar en *mailbox* creado con desbordamiento de *buffer* y 4. Obtención de la bandera.

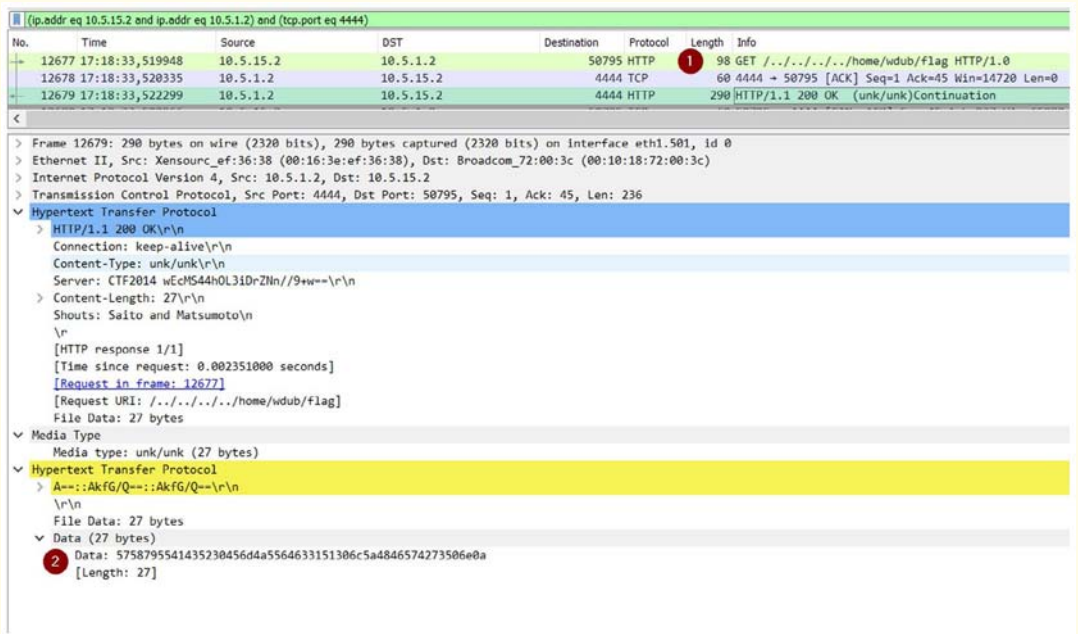
Figura 8.
10.5.1.110 ataca el servicio IMAP de 10.5.9.2



Fuente: elaboración propia. Captura de imagen de Wireshark.

En la Figura 9 se puede observar la obtención de la bandera mediante un ataque de "*directory traversal*" (también conocido como "salto de directorio", "cruce de directorio" o "*path traversal*"), que consiste en escapar del directorio web en que el servicio debería estar confinado. En este caso, la bandera se encuentra dentro del archivo /home/wdub/flag.

Figura 9.
Obtención de bandera usando (Sid:1113, "WEB-MISC http directory traversal")



Fuente: elaboración propia. Captura de imagen de Wireshark.

Debido a la componente de defensa del CTF, los ataques que son exitosos al principio pueden dejar de serlo posteriormente, dependiendo principalmente de la calidad de los parches binarios creados por los defensores. Además, existen otros ataques, algunos de los cuales son exploratorios, mientras que otros no están claros si fueron o no exitosos. También se registran ataques entre estaciones de trabajo, los cuales no otorgan puntos en el CTF, pero pueden ayudar a entorpecer el trabajo de un equipo contrincante

Aplicación del método a la DEF CON 22 CTF

Tras el procesamiento de las capturas de red del CTF mediante Snort, se obtuvo un archivo consolidado de alertas que se filtró según el atacante, el objetivo, o por ambos. Posteriormente, estas alertas filtradas se sometieron al algoritmo de clasificación según la CKC. A continuación, se presentan dos casos representativos para caracterizar un CTF:

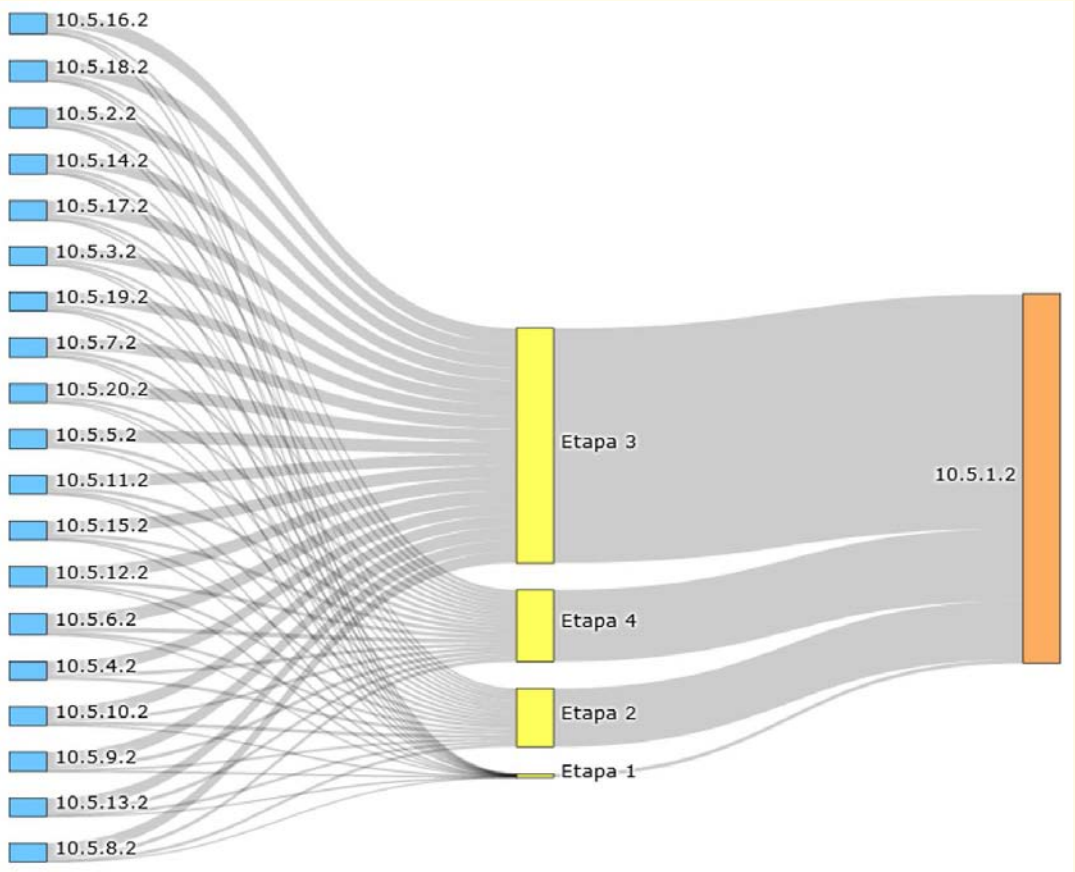
i) Todos los equipos contra el equipo PPP (10.5.1.2).

ii) El equipo HITCON (10.5.9.2) atacando al equipo PPP (10.5.1.2).

i) Escenario de ataque usando 10.5.1.2 como objetivo

En la Figura 10 se pueden apreciar los ataques dirigidos al servidor 10.5.1.2, el cual es defendido por el equipo PPP. Estos ataques son realizados por los 19 equipos restantes. Los anchos de cada una de las líneas son similares entre sí para el caso de las que ingresan a la misma barra, que representa cada etapa del CKC, por lo que a primera vista parecen similares entre sí.

Figura 10.
Cada uno de los 19 equipos pasa por las cuatro etapas del CKC para atacar al objetivo defendido por el equipo PPP, 10.5.1.2

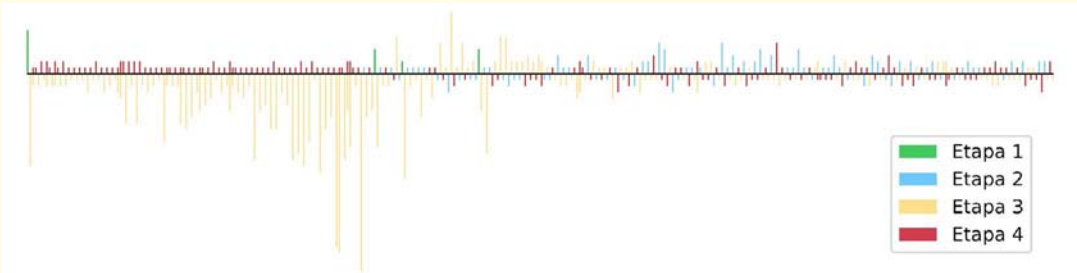


Fuente: elaboración propia. Generado automáticamente con el software MSNA.

ii) Escenario de ataque usando 10.5.1.2 como objetivo y 10.5.9.2 como atacante

Durante el CTF se registraron 995 ataques desde 10.5.9.2 a 10.5.1.2, representados en la Figura 11. Estos se inician con siete ataques de etapa 1, seguidos de 15 de etapa 3. Desde la tercera línea, se puede apreciar que esta es roja, lo que indica que es un ataque de etapa 4. Luego de esto, se mantiene una intermitencia principalmente compuesta por varios ataques de etapa 3 y unos pocos de etapa 4. Solo ocasionalmente aparecerán otras etapas. La línea más larga corresponde a 32 ataques tipo etapa 3 del CKC.

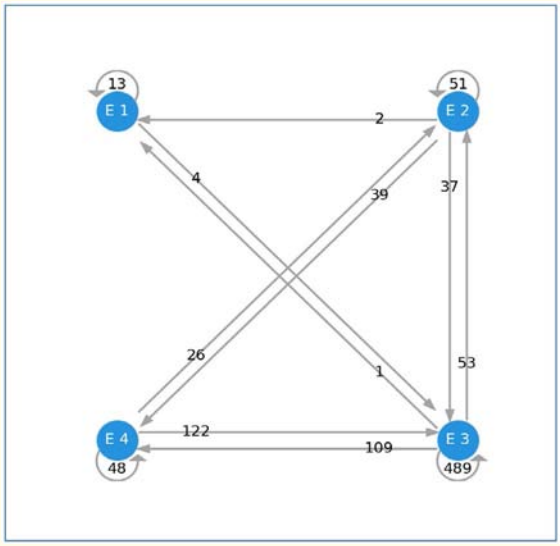
Figura 11.
Línea de tiempo del ataque de 10.5.9.2 considerando los cambios de etapa de CKC



Fuente: elaboración propia. Generado automáticamente con el software MSNA.

En la Figura 12, los ataques de una etapa particular del CKC corresponden al mismo número de etapa. Se puede apreciar que la mayoría de los estados son 3, cuyo estado previo era 3, y que luego, por orden de cantidad, están los estados 3 cuyo estado previo era 4, y los de estado 4 donde el estado previo era 3. Los estados relacionados con 1 y 2 son menores que los otros estados, por lo que parece que hubo poca actividad relacionada con estas etapas.

Figura 12.
Máquina de estados para los ataques de 10.5.9.2 dirigidos a 10.5.1.2 en donde los valores de los estados son las etapas de CKC



Fuente: elaboración propia. Generado automáticamente con el software MSNA.

Discusión de los resultados DEF CON 22 CTF

i) Escenario de ataque usando 10.5.1.2 como objetivo

En la Figura 10, la similitud en los anchos de cada una de las líneas que ingresan a la

misma etapa CKC implica similitud en la cantidad, proporción y clasificación de los ataques entre los diferentes atacantes. Considerando que los atacantes pueden hacer ingeniería inversa con los ataques recibidos, es esperable que los equipos se retroalimenten entre sí en la medida en que son atacados. Esta similitud permite utilizar un solo ataque como referencia para describir al resto.

ii) Escenario de ataque usando 10.5.1.2 como objetivo y 10.5.9.2 como atacante

En cuanto al gráfico de máquina de estado, en la Figura 12 se puede apreciar que el estado que tiene más alertas es el de etapa 3, con un total de 652, de los cuales 489 tienen como etapa previa la misma etapa 3 y los restantes de otras etapas. El mayor número de ataques de etapa 3 puede deberse a que hay que hacer múltiples ataques de esta etapa antes de lograr un ataque exitoso. Luego de lograr el éxito, se realizan las exfiltraciones de información o, en este contexto, la obtención de la bandera mediante un ataque de etapa 4. Después de esto, se inicia nuevamente el ciclo del MSNA. El motivo para iniciar un nuevo MSNA es que el valor de la bandera cambia en cada round, generando nuevos puntos cuando se tiene la bandera actualizada, también podrían haberse parchado servicios de un round a otro, por lo que se necesita buscar, detectar y explotar nuevas vulnerabilidades. El evento DEF CON 22 CTF se dividió en 272 rondas de cinco minutos cada una. Al comienzo a cada equipo se le asignaron 2502 banderas, las que se distribuyeron igualmente entre los seis servicios[15].

En cuanto a la información obtenida al inicio, debería ser suficiente para toda la operación, dado que no se producen cambios en la infraestructura de red entre un MSNA y otro. Por lo tanto, no es necesario volver a obtenerla.

Al revisar las figuras 11 y 12 se puede observar que, una vez finalizado un ciclo de CKC o de un ataque MSNA, estos ciclos se reinician. Se pueden identificar ciclos cortos que solo regresan a ataques de etapa 3 y otros ciclos un poco más largos que regresan hasta la etapa 2. La falta de pasos de etapa 1 puede deberse a que no se necesita un nuevo descubrimiento de la red. En total, se completaron 148 ataques MSNA, de los cuales 26 provienen de la etapa 2 y 122 provienen de la etapa 3.

CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se caracterizaron los ataques MSNA en un ejercicio CTF mediante el enfoque forense. El método usado para esto fue un KCS2ME, el cual se apoyó en una herramienta de amplio uso y de muchos años de desarrollo como Snort. De esa manera, se aplicó un conjunto de reglas para determinar las cualidades o rasgos característicos de los ataques del DEF CON 22 CTF. Si bien un enfoque basado en reglas es considerado rígido, la clasificación de los ataques individuales en etapas CKC lo hace más general, pudiendo con ello haber hecho una comparación entre ataques MSNA.

Claramente, existen diferencias entre un CTF de ataque-defensa y un MSNA en ambiente real. Por ejemplo, 1) el sistema a defender es prácticamente idéntico al sistema que se debe atacar, y en un ataque real solo se podría acceder al *software* base, como servicios y sistema operativo, pero no a configuraciones ni *software* hecho a medida, como páginas

15 Ibíd.

web. 2) Es posible descubrir vulnerabilidades y ataques explotables al hacer ingeniería inversa de los ataques recibidos. 3) En el mundo real, un atacante externo necesita obtener información de la red mediante la realización de ataques que se encuentran clasificados en la etapa 1 del CKC, mientras que, en un CTF, las redes y los blancos de los ataques están predefinidos. 4) Una vez que se realiza un ataque exitoso, no pasa mucho tiempo para que el equipo de defensa pueda crear una contramedida en forma de parche binario, lo que obliga al atacante a crear un nuevo ataque o una nueva variante del anterior.

Futuros investigadores podrían abordar las limitaciones de este estudio de varias maneras. Por ejemplo, podrían ampliar la cantidad de reglas clasificadas, las cuales hoy son 7750; usar otros CKC para buscar patrones de ataque generales diferentes, como ATP, a fin de complementar el actual módulo. También podrían buscar formas de encontrar patrones de CKC en distintas combinaciones de orígenes y destinos de las alertas, así como también su extensión a casos reales. También se podrían considerar la colusión entre equipos, en donde los ataques a estudiar no son solo uno a uno, o bien un estudio que se enfoque dentro de cada round.

REFERENCIAS BIBLIOGRÁFICAS

BUKAC, Vit, LORENC, Vaclav y MATYÁŠ, Vashek. 2014. Red Queen 's Race: APT Win-Win Game [en línea]. En: CHRISTIANSON, B., MALCOLM, J., MATYÁŠ, V., ŠVENDA, P., STAJANO, F. y ANDERSON, J. Security Protocols XXII. *Security Protocols 2014. Lecture Notes in Computer Science*. Suiza: Springer, Cham, 55-61. Disponible en: https://doi.org/10.1007/978-3-319-12400-1_7 [Consulta: 29 de agosto de 2021].

HAINES, Joshua. 2000 Darpa intrusión detection scenario specific data sets [en línea]. 2020. Disponible en: https://archive.ll.mit.edu/ideval/data/2000data.html [Consulta: 29 de agosto de 2021].

HUTCHINS, Eric, CLOPPERT, Michael y AMIN, Rohan. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research [en línea], vol. 1, no. 1, 1-14. Disponible en: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf [Consulta: 29 de agosto de 2021].

International Organization for Standardization [ISO]. ISO/IEC. 27000:2016. Information technology — Security techniques — Information security management systems — Overview and vocabulary [en línea]. 2016. Disponible en: https://www.iso.org/standard/66435.html [Consulta: 29 de agosto de 2021].

JULISCH, Klaus. 2003. Clustering intrusion detection alarms to support root cause analysis. ACM Transactions on Information and System Security [en línea], vol. 6, no. 4, 443-471. Disponible en: https://doi.org/10.1145/950191.950192 [Consulta: 29 de agosto de 2021].

KUCEK, S. y LEITNER, M., 2020. An empirical survey of functions and configurations of open-source capture the flag (CTF) environments. Journal of network and computer applications [en línea], vol. 151, no. 102470, ISSN 1084-8045. DOI 10.1016/j.jnca.2019.102470. Disponible en: http://dx.doi.org/10.1016/j.jnca.2019.102470.

MENDES, J. y SOARES, R. 2019. Flexible Approach to Multi-Stage Network Attack Recognition. International Journal of Computer Science and Information Security (IJCSIS) [en línea], vol. 17, no. 8, 67-73. Disponible en: https://www.academia.edu/40458556/Flexible_Approach_to_Multi_Stage_Network_Attack_Recognition [Consulta: 29 de agosto de 2021].

MIRKOVIC, J. y PETERSON, P. 2014. Class Capture-the-Flag Exercises. En: 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14) [en línea]. San Diego, California, agosto de 2014. Disponible en: https://www.usenix.org/biblio/class-capture-flag-exercises [Consulta: 29 de agosto de 2021].

NAVARRO, Julio, DERUYVER, Aline y PARREND, Pierre. 2018. A systematic survey on multi-step attack detection. Computers Security [en línea], vol. 76, 214-249. Disponible en: https://doi.org/10.1016/j.cose.2018.03.001 [Consulta: 29 de agosto de 2021].

POLS, Paul. 2022. The Unified Kill Chain [en línea]. Disponible en: https://www.unifiedkill-chain.com/assets/The-Unified-Kill-Chain.pdf [Consulta: 24 de mayo de 2023].

SINGH, Vivek, CALLUPE, Steven y GOVINDARASU, Manimaran. Testbed-based Evaluation of SIEM Tool for Cyber Kill Chain Model in Power Grid SCADA System. En: 2019 North American Power Symposium (NAPS) [en línea]. Kansas, Estados Unidos, octubre de 2019, pp. 1-6. Disponible en: https://www.researchgate.net/publication/336623025_Testbed-based_Evaluation_of_SIEM_Tool_for_Cyber_Kill_Chain_Model_in_Power_Grid_SCADA_System [Consulta: 29 de agosto de 2021].

ŠVÁBENSKÝ, V., ČELEDA, P., VYKOPAL, J. y BRIŠÁKOVÁ, S., 2021. Cybersecurity knowledge and skills taught in capture the flag challenges. Computers & security [en línea], vol. 102, no. 102154, ISSN 0167-4048. DOI 10.1016/j.cose.2020.102154. Disponible en: https://www.sciencedirect.com/science/article/pii/S0167404820304272 [Consulta: 24 de mayo de 2023]

University of New Brunswick [UNB]. CSE-CIC-IDS2018 on AWS. A collaborative project between the Communications Security Establishment (CSE) & the Canadian Institute for Cybersecurity (CIC) [en línea]. 2020a. Disponible en: https://www.unb.ca/cic/datasets/ids-2018.html [Consulta: 29 de agosto de 2021].

University of New Brunswick [UNB]. Intrusion detection evaluation dataset (CIC-IDS2017) [en línea]. 2020b. Disponible en: https://www.unb.ca/cic/datasets/ids-2017.html [Consulta: 29 de agosto de 2021].

WILKENS, Florian, ORTMANN, Felix, HAAS, Steffen, VALLENTIN, Matthias y FISCHER, Mathias. Multi-Stage Attack Detection via Kill Chain State Machines [en línea]. 2021. Disponible en: https://arxiv.org/abs/2103.14628 [Consulta: 29 de agosto de 2021].

YAM, W.K.J., 2016. *Strategies used in capture-the-flag events contributing to team performance*. S.l.: Monterey, California: Naval Postgraduate School. [en línea]. Disponible en: https://hdl.handle.net/10945/48498 [Consulta: 23 de mayo de 2023].