https://doi.org/10.22201/dgtic.ctud.2023.1.1.17

Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 169 - 184

Implementación de técnicas de observabilidad en el Centro de Monitoreo de la Red

Información del reporte:

Para citar este reporte técnico:

Ramírez Fernández, E. R. (2023). Implementación de técnicas de observabilidad en el Centro de Monitoreo de la Red. *Cuadernos Técnicos Universitarios de la DGTIC, 1* (1), páginas (169 - 184). https://doi.org/10.22201/dgtic.ctud.2023.1.1.17

Esteban Roberto Ramírez Fernández

Dirección General de Cómputo y de Tecnologías de Información y Comunicación, Universidad Nacional Autónoma de México

esteban.ramirez@unam.mx

ORCID: 0000-0002-2169-6233

Resumen:

El presente reporte muestra el resultado de la evaluación de aplicaciones orientadas al análisis de información, para conformar una arquitectura de monitoreo que contribuya al análisis de eventos que afectan la operación de la red de datos al interior de la UNAM operada por el Centro de Monitoreo de la Red (*NOC*) de RedUNAM. Se incorpora el concepto de "observabilidad" que permite incrementar la visibilidad a partir de la concentración de información proveniente de bitácoras y sistemas de monitoreo, entre otros, para lo cual se considera cualquier fuente de información relevante para la operación de las redes, y se crean tableros de control con gráficas que permiten la consulta en tiempo real de los datos recolectados. La integración de software emplea aplicaciones de monitoreo activo, pasivo, e información de protocolos de comunicación para el análisis de incidentes en la red. El desarrollo se enfoca en la problemática de identificación y solución de incidentes de comunicaciones en los enlaces de la red de datos de la UNAM, para lo que se concentran datos de *hardware*, *software* y de aplicaciones operativas que se comunican al interior de la UNAM y a Internet, y así atender la responsabilidad del *NOC* de RedUNAM de mantener la conectividad a Internet y redes académicas en óptimas condiciones.

Palabras clave:

Sistemas de monitoreo, técnicas de observabilidad, incidentes en redes de datos.

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 170 - 184

1. INTRODUCCIÓN

La implementación de mecanismos de monitoreo para conocer el estado de operación de las redes de datos es una actividad clave debido a la importancia de conocer la disponibilidad de los enlaces de comunicaciones que influyen de forma directa en la percepción de uso de los servicios de Tecnologías de Información (*TI*); estos mecanismos han evolucionado impulsados por la migración de procesos críticos a sistemas informáticos. Se requiere de una detección inmediata de los eventos que afectan a cualquier nivel las comunicaciones, si se toma en cuenta que las redes se componen de infraestructura física y virtual y se necesita obtener información de todos los elementos involucrados en una falla.

En el Centro de Monitoreo de la Red (*NOC*) de RedUNAM de la DGTIC se busca constantemente mejorar la visibilidad de las conexiones de red debido a que el *NOC* es responsable de vigilar su operación y garantizar la comunicación a las entidades y dependencias universitarias por medio de enlaces dedicados al campus central de la UNAM con apoyo de proveedores de telecomunicaciones en el área metropolitana y en todo el territorio de México, tanto para las conexiones a Internet como a través de redes académicas. Contar con información completa de las comunicaciones desde que se originan por el cliente hasta ser atendidas por un servidor y a lo largo de todo el trayecto involucra soluciones tecnológicas en constante evolución. Esta complejidad, donde confluyen diferentes elementos de comunicación físicos y virtuales, motiva al cambio constante en la atención de incidentes de red.

Hasta ahora el diagnóstico de incidentes ha empleado la información operativa que ofrecen todos los dispositivos mediante el protocolo simple de administración de red (*SNMP, Simple Network Management Protocol*) y la tecnología llamada Netflow (propietaria de la marca CISCO para análisis de flujos de red), así como el estándar IPFIX para la misma tarea.

Buscar la causa raíz de incidentes de red para diagnosticar y resolverlos de forma ágil implica una estrategia para la recolección de información de la arquitectura de la red de la UNAM y datos sobre las aplicaciones que hacen uso de ella en centros de datos y servidores dedicados, así como los protocolos de comunicación de red, para estar en posibilidad de atender fallas de comunicación y detectar de manera rápida, por ejemplo, los problemas de mal funcionamiento físico.

Para apoyar esta tarea se han explorado técnicas que emergen del mundo de la operación de servicios en la nube y desarrollo de *software*. Entre ellas se destaca el concepto "*Observabilidad*"[1], el cual está relacionado con la capacidad de obtener datos de distintas fuentes y formatos que sean de interés, para proveer de un panorama completo de información que en este caso se aplica con un enfoque hacia la solución de incidentes de red.

[1] Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara and Stephen Thorne. "The Site Reliability Workbook" Google. Recuperado el 1 de noviembre de 2023, de https://sre.google/workbook/table-of-contents/ (SER), editado por de la compañía Google.

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 171 - 184

2. OBJETIVO

2.1 OBJETIVO GENERAL

Generar una arquitectura compuesta por soluciones de software que permita la implementación de estrategias para el análisis de información de diferentes fuentes de datos, obtenidos de equipos y aplicaciones que operan en la red de datos de la UNAM para la mejora en la atención de incidentes de red.

2.2 OBJETIVOS ESPECÍFICOS

Conjuntar datos para analizar el comportamiento de eventos asociados a las comunicaciones, para identificar incidentes, establecer su causa y permitir darles solución inmediata y registrar la causa raíz de problemas frecuentes.

Integrar *software* y herramientas que permitan concentrar información de eventos (alertas de monitoreo) como resultado del monitoreo tradicional, por medio de *SNMP* de red y el análisis de bitácoras de equipos para relacionar y concentrar la información en un solo punto.

Generar tableros de control y reportes con apoyo de fuentes de datos concentradas, con apoyo de funciones de información que permitan hacer filtros para presentar el análisis de comportamientos anormales en un formato útil tanto para personal operativo como directivo.

3. DESARROLLO

Para el desarrollo de este reporte técnico se consideró importante contextualizar el origen de la iniciativa, por lo que a continuación se abordan los antecedentes y posteriormente se describe la metodología empleada para la implementación de las pruebas de *software* con apoyo de tableros de control.

3.1. ANTECEDENTES

El *NOC* de RedUNAM es el área responsable de operar y monitorear servicios de comunicaciones (enlaces de datos) que concentran diferentes equipos de red y tecnologías de comunicación, cuyo correcto funcionamiento contribuye a determinar el nivel de disponibilidad de la red de datos de la UNAM.

Para mejorar la tarea de monitoreo de enlaces de área amplia (*WAN*, *Wide-Area Network*) y así mantener la disponibilidad de los enlaces de datos en procesos cada vez más críticos tales como exámenes y procesos administrativos que operan sobre estas redes de datos, se volvió aún más importante contar con información del estado de todos los elementos que conforman los servicios de comunicación, en tiempo real.

Para incrementar la capacidad del monitoreo al ampliar la visibilidad y las relaciones de los datos disponibles, se exploraron alternativas al esquema tradicional que opera solicitando el estado de un elemento cada determinado tiempo. Este proceso fue realizado en el periodo de mayo a octubre de 2021, en el que se probaron mecanismos de monitoreo y manejo de la información alternativos al funcionamiento del protocolo *SNMP*.

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 172 - 184

Una motivación principal fue complementar la información consultable por medio de un agente SNMP, debido a que si un elemento no cuenta con soporte de este protocolo, pierde la capacidad de ser monitoreado y es únicamente compensado con la experiencia de los operadores en eventos previos. Este es el caso de las aplicaciones que operan sobre las redes de datos para las comunicaciones, como redes virtuales o las propias aplicaciones que ofrecen servicios en red como son los servicios web. Sin embargo, el protocolo sigue siendo la principal fuente de información de los equipos de red.

Para resaltar las técnicas de monitoreo exploradas y compararlas con las tradicionales, se agrupan a continuación las técnicas empleadas; se toma como referencia su enfoque de consulta (monitoreo activo) o de escucha, así como la notificación de un evento (monitoreo pasivo), como sigue:

Técnicas de Monitoreo tradicional empleadas en el *NOC* de RedUNAM

Monitoreo activo de incidentes:

- Monitoreo que resulta de la recolección de información de estado de los enlaces por medio de protocolos *ICMP* y *SNMP*.

Monitoreo pasivo de incidentes:

- Recolección de información por medio del protocolo *SNMP*, a través del uso de la función "*traps*" previamente configurada en los equipos de red a monitorear.
- Monitoreo de bitácoras de equipos por medio del protocolo *Syslog* definido por la *IETF* en los documentos *Request For Comments* 5424, 426 y 6587.
- Implementación de *RMON* (*Remote Network Monitoring*) para monitoreo del estado de elementos que soportan esta tecnología.

Técnicas exploradas y actualmente empleadas para el diagnóstico de incidentes de red:

Monitoreo pasivo de incidentes

- Monitoreo de bitácoras de los equipos de red y aplicaciones concentradas en bases de datos de series de tiempo.
- Monitoreo pasivo de protocolo *BGP* para la detección de cambios de rutas; concentra eventos para ayudar a identificar caídas de rutas lógicas.
- Monitoreo basado en trazas: Rastreo de los eventos a través de una trayectoria en las redes de datos.

Además de las técnicas de monitoreo, se exploraron técnicas de almacenamiento que permitieron ampliar las posibilidades de concentración de información en un solo punto.

Se han explorado al momento estrategias de almacenamiento del tipo binario, Round Robin y relacional, para conjuntar los diferentes mensajes en una sola base con capacidad suficiente para la retención, lectura y escritura requerida (con un mínimo de cinco años).

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 173 - 184

3.2. PROBLEMÁTICA POR RESOLVER

La complejidad de los sistemas de comunicaciones ha ampliado las fuentes de información, desde la infraestructura hasta el software que realiza las comunicaciones. La estrategia más común aplicada al interior del *NOC* de RedUNAM para el diagnóstico de un incidente ha sido encontrar la relación de la información del monitoreo de un incidente proveniente de estas fuentes de información, por medio de los distintos sistemas de apoyo, comparar los resultados del diagnóstico y concluir con base en la experiencia técnica del operador la causa de un incidente. Esta actividad cuenta con diferentes tipos de fuentes de información (enlaces, servidores, protocolos y aplicaciones) consultados de forma aislada; para el diagnóstico se involucran los siguientes elementos:

- Enlaces de datos (compuestos de equipos, interfaces y medios de comunicación).
- Servidores que ofrecen servicios (monitoreo de valores operativos de los mismos).
- Protocolos de comunicación (si se cuenta con información de estados y comunicación).
- Datos de aplicaciones (monitoreadas bajo solicitud).

Cada elemento genera información con un formato distinto que se consulta de forma independiente para el diagnóstico de un evento con posible afectación al interior de la red de datos de la UNAM, a la que se agrega información valiosa que tal vez no ha sido identificada porque se analiza en su propio entorno.

Encontrar las relaciones en el proceso de análisis de los datos anteriormente mencionados refleja la importancia de integrar nuevas técnicas que presenten reportes gráficos de la información recolectada por distintos mecanismos, para facilitar la identificación visual de un evento y apoyar la toma de decisiones acerca de un incidente de red por parte del personal especializado y directivos.

3.3. FUNDAMENTOS TEÓRICOS

El término "*Observabilidad*"[2] fue desarrollado para contar con nuevas formas de mantener vigilada la operación de los elementos virtuales que conforman los servicios de TI en la nube.

Anya Bragin (2019) menciona que "la observabilidad no es algo que un proveedor entrega en una caja, es un atributo de un sistema que creas, similar a la facilidad de uso, la alta disponibilidad y la estabilidad". El objetivo de diseñar y crear un sistema "observable" es asegurarse de que cuando se ejecute en producción, los operadores responsables puedan detectar comportamientos no deseados y tengan información procesable para localizar la causa raíz de manera eficaz.

Con base en este razonamiento, se desarrolla la presente propuesta con un enfoque de observabilidad a las redes de datos para apoyar la necesidad de integración de información relativa a eventos operativos.

A partir de este término, se buscó implementar este enfoque centrado en las comunicaciones de las redes de datos, ya que como se mencionó, originalmente surge de entornos de desarrollo de *software* para tecnologías de información de la nube, donde todos los elementos son virtuales y la capacidad de ser observados se genera a partir de la unificación de los eventos recolectados e indicadores de cada aplicación.

[2] Se rastrea el origen del término "observabilidad" del libro S.R.E.(2018), donde Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara y Stephen Thorne al libro "*The Site Reliaability Workbook*" (SRE) (2018) editado por la compañía *Google,* mencionan los principios relacionados con lo que se conoce actualmente como observabilidad.

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 174 - 184

Para la empresa de soluciones de telecomunicaciones *CISCO* (s.f), en su portal comercial se menciona que: "la observabilidad es un proceso que utiliza herramientas de *software* para detectar problemas mediante la observación de las entradas y las salidas de la oferta tecnológica". En el mismo artículo se menciona que "las herramientas de observabilidad recopilan y analizan una amplia gama de datos, incluido el estado y el rendimiento de las aplicaciones, las métricas de negocios como las tasas de conversión, la asignación de la experiencia del usuario, y la telemetría de la infraestructura y la red para resolver problemas antes de que afecten a los *KPI* empresariales".

4. METODOLOGÍA APLICADA

Siendo relevante para el análisis de información la selección de *software* para el análisis de datos, la cantidad de espacio requerido para concentrar la información y la generación de una propuesta que otorgue valor a la operación del monitoreo de la red de la UNAM, se dividió la metodología en cuatro etapas proponiendo una estrategia donde se evaluaron diferentes técnicas de monitoreo y se conformó una arquitectura operativa para explotar la información en el proceso actual de análisis de incidentes de red.

En el desarrollo de estas etapas para la selección de *software* y análisis de información se emplearon dos estrategias que contribuyeron a definir arquitectura y evaluar qué información contribuye a la solución de incidentes de red. Se eligió como método de selección de *software* al método ***MoSCoW***, complementado con el uso de los métodos **inductivo y deductivo** para centrar la elección de las capacidades consideradas como deseables en las soluciones de *software* y la identificación de información para la solución de incidentes de red considerando la presentación, flexibilidad y navegación en tiempo real de los datos.

Método *MoSCoW*:

El método para la selección de *software MoSCoW* es un mecanismo empleado como parte de las mejores prácticas de *TI* provistas por la biblioteca de buenas prácticas *ITIL* en su versión 4 (2019) .

MoSCoW permite separar los requisitos o características en cuatro categorías:

- M (*MUST*) La solución tiene que cumplir la capacidad. Si no se cumple se verá comprometido el éxito del proyecto.
- S (*SHOULD*) La solución debería cumplir este requisito siempre que sea posible. El éxito del proyecto no depende del cumplimiento, aunque es de alta prioridad.
- C (*COULD*) Sería relevante incluir la capacidad. Es una característica opcional.
- W (*WON'T*) Estos requisitos no se consideran, aunque podrán incluirse a futuro.

En complemento, los métodos inductivo y deductivo para selección de *software* de acuerdo con (Rodríguez Jiménez, 2017), son métodos científicos de indagación y de construcción del conocimiento, y resulta útil su empleo para la generación de criterios que permitan establecer diferencias entre varias opciones de *software*, en el caso que nos ocupa.

Se hace uso del método *inductivo* en la selección de *software* por medio de la observación de los tipos de soluciones más reconocidos, mencionados en estudios de empresas reconocidas como *Gartner*, identificando patrones de operación de acuerdo con las características ofrecidas.

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 175 - 184

En cambio, el método *deductivo* apoya el desarrollo de los criterios de selección con base en las capacidades que se buscan para integrar una arquitectura que ofrezca características similares, permitiendo la selección de *software* final derivado del análisis de los criterios generados.

4.1. ETAPA 1: ANÁLISIS DE LA INFORMACIÓN

En la primera etapa, para delimitar el alcance de una propuesta de *software* se analizó el tipo de información disponible partiendo de la información enviada por los equipos de red:

Concentrar y analizar la información

El análisis requiere concentrar la información, para ello se evaluaron las fuentes de información operativa física y virtual importantes para una comunicación completa, a través de una red de datos, identificando los siguientes elementos de comunicación:

- Origen de la comunicación (cliente y aplicación que puede consumir otra aplicación).
- Medios de transmisión que emplean (sirve para identificar fuentes de información).
- Equipos de red involucrados (sirve para definir tecnologías de monitoreo disponibles).
- Destino de la comunicación (unidireccional, bidireccional) y protocolos empleados.

Fuentes de información:

La aplicación completa del concepto de observabilidad implica obtener información de todos los elementos que componen un servicio; sin embargo, centrándose en el alcance de esta propuesta, se busca recuperar información de todos los elementos que ofrezcan datos del proceso de las telecomunicaciones, para ello se identifica cualquier proceso que sea parte de una comunicación; para ello se emplean protocolos *SNMP*, *ICMP*, información de flujos de red, bitácoras de información, excepciones de aplicación y cualquier fuente que brinde datos relevantes.

Derivado del análisis de la etapa 1, se consideran:

- Fuentes de información (protocolos, bitácoras y alertas disponibles)
  - *NETFLOW, IPFIX, SYSLOG, SNMP.*
- Interfaces para la consulta (cuáles serían los medios para concentrar la información)
  - Puertos lógicos de *SNMP* (161 y 162 *UDP*) y *SYSLOG* (514 *TCP*).
- Métodos de recolección (pasiva y activa)
  - *SYSLOG* y *SNMP.*
- Almacenamiento de la información (tipo de bases de datos y cómo se relacionarían)
  - Bases de datos basadas en series de tiempo.

4.2. ETAPA 2: EVALUACIÓN DE SOLUCIONES DE SOFTWARE

Para reducir el tiempo de evaluación de soluciones de monitoreo y observabilidad dentro de la gran cantidad de soluciones que ofrecen características de valor para el diagnóstico y solución de incidentes de red, se consideró como punto de comparación el análisis de la empresa referente de estudios de *software* propietario *Gartner.*

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 176 - 184

Considerando las características de las soluciones de *software* propietarias que son de gran valor en las tareas de concentración y análisis de información, destacan las compañías ***Elastic, New Relic, Datadog, Splunk*** y ***Dynatrace***. Sin embargo, derivado de una evaluación inicial del costo de uso de los productos con información de las páginas oficiales, extrapolando el costo por *gigabyte* de información procesado, se consideraron no costeables, pero sirven de base para elaborar una arquitectura compuesta de soluciones de software libre integrando capacidades destacadas de las propietarias.

La evaluación de características que se identificaron en el software propietario disponibles en *software* libre y aportan valor a las necesidades del *NOC* de RedUNAM son:

- Soporte para base de datos libre de alta velocidad de lectura y escritura,
- Interfaz o mecanismos de conexión con fuentes de información externas, y
- Tableros de control web con facilidad de uso para los usuarios expertos y no expertos.

El listado detallado se encuentra en el **Anexo A, tabla 1: Evaluación *MoSCoW***.

Las características antes mencionadas posteriormente se evaluaron en las soluciones con mayor comunidad de soporte realizando la selección de entre las aplicaciones listadas, destacando: *Grafana*, *Loki, PNP, Telegraf* y *Rsyslog*. En el **Anexo A, tabla 2: Selección de Software** se ubican también las aplicaciones evaluadas.

4.3. ETAPA 3: ALMACENAMIENTO DE LA INFORMACIÓN

Durante la evaluación del *software* para cumplir con las necesidades de almacenamiento se buscó conformar una arquitectura con bases de datos no propietarias, puesto que la arquitectura se compone de varias fuentes de datos y se pretende que concentre gran cantidad de la información, además de permitir la consulta de información en tiempo real para generar tableros.

Esta tarea, con colectores de datos propietarios, requiere muchos recursos y licencias. Por esta razón se eligieron colectores de información basados en series de tiempo que son los incluidos en la evaluación anterior.

Para esta tarea se considera la implementación de más de un colector de información para poder incorporar la mayor cantidad de datos posible y posteriormente poderla relacionar en tableros de control. Los colectores de información en forma de eventos con marcas de tiempo seleccionados fueron:

- *FluentD*
- *Rsyslog*
- *PNP NAGIOS*
- *LOKI*

Se le llaman colectores debido a la función que realizan de captura, almacenamiento y posibilidad de consulta de información guardada; cada *software*, como el caso de *FluentD*, es un sistema que genera su estructura para gestionar la información recuperada, habilitando interfaces de comunicación y también interfaces de usuario (*UI*) para la administración de la base de información en su conjunto. La ventaja de usar este tipo de colectores proviene de que permiten el registro de cada evento empleando una marca de tiempo.

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 177 - 184

4.4. ETAPA 4: DEFINICIÓN DE ARQUITECTURA

Derivado del análisis de aplicaciones, se generó una arquitectura compuesta de aplicaciones libres para alimentar el *software* elegido para tableros de control *Grafana*, cuyo mayor diferenciador fue la capacidad de integrar como fuentes de datos al sistema de monitoreo de disponibilidad, a través de interfaces.

La arquitectura se estructuró como sigue:

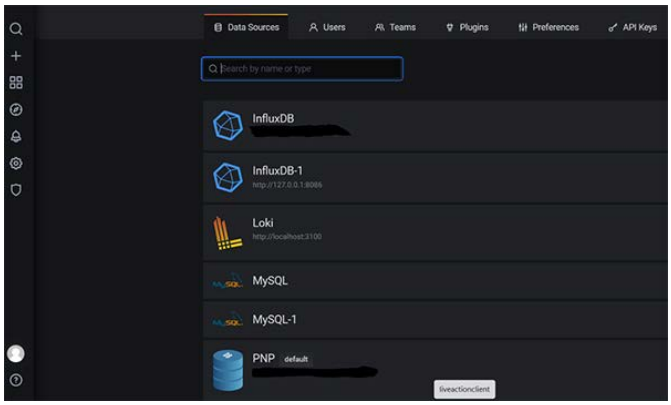
- Sistema operativo *Linux Base Debian*.
- Implementación de *software PNP4Nagios* (para obtener el dato de monitoreo SNMP).
- Implementación de *FluentD* (concentrar información hacia base de datos *Json*).
- *Software* de recolección de eventos *RSYSLOG*.
    - Duplicación de bitácoras seleccionadas hacia el *software FluentD*.
    - Configuración de equipos remotos para envío de bitácoras *SYSLOG*.
- Implementación de *software LOKI* (filtrado y consultas desde bitácoras).
- Manejador de tableros de control *Grafana*.
    - Fuentes de datos integradas para recuperación de información.
        - Fuente *LOKI*.
        - Fuente *FluentD*.
- Fuente *PNP for Nagios Firewall* local integrado (*IPfilter*).
- Servidor de correo para liberar alertas *Postfix* (*para envío de alertas*).

5. RESULTADOS

Se logró la implementación y vinculación del sistema de monitoreo *Nagios* permitiendo consultar el estatus de los servicios monitoreados a nivel de disponibilidad por medio de la interfaz *PNP 4 Nagios* a *Grafana*; lo anterior se muestra en la figura 1, con las fuentes de datos configuradas en el *software Grafana*, en la arquitectura implementada para la evaluación realizada.

Figura 1

Fuentes de datos integradas al sistema Grafana



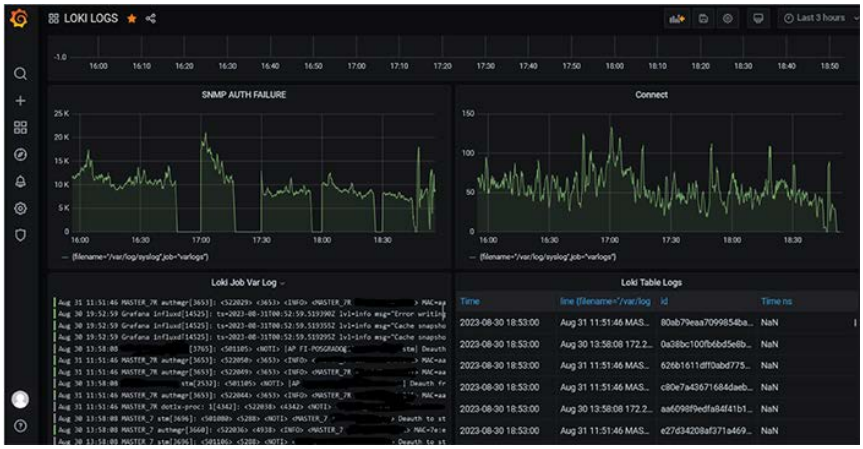
https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 178 - 184

Se envía información de los servicios monitoreados de acceso a Internet y monitoreo de protocolos de red que podría apoyar a la identificación de incidentes.

Se implementó el servicio de captura de bitácoras, primero por medio de la aplicación *Rsyslog* y su posterior replicación a una segunda aplicación para su tratamiento, *FluentD*, permitiendo la consulta desde *Grafana* por medio de la solución *Loki* para generar tableros de control. Se implementaron las interfaces para el acceso a información de monitoreo *PNP4Nagios* y *LOKI*, como se muestra en la figura 2.

Figura 2

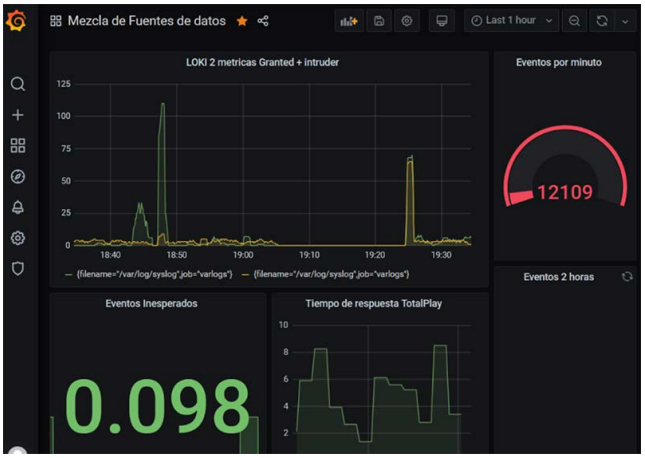
Consulta de información y gráfica de conteo de eventos con Grafana + Loki



También se configuraron tableros de control con la información de las fuentes de datos para concentrarlos en una misma vista y comparar información de distintas fuentes, como se ve en la figura 3, con un ejemplo de la contabilización de eventos de diferentes fuentes de datos (*Nagios + PNP + Grafana, RSYSLOG + LOKI + Grafana*) y el cambio de rango a minuto y día.

Figura 3

Conteo de bitácoras con la palabra "intruder" con Loki + tiempos de respuesta a Internet



https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 179 - 184

Se lograron generar tableros de control en *Grafana*:

- Gráficas de los valores de respuesta consultados con el sistema *Nagios* + *PNP4Nagios*.
- Comparación en una sola gráfica de hasta tres elementos monitoreados en *Nagios*.
- Creación de umbrales para monitoreo de patrones que salen de lo que se espera.

Se hicieron filtros en los tableros de control de *Grafana* + *Loki*:

- Se obtuvo un reporte gráfico por medio de *Grafana* + *LOKI* sobre las bitácoras recolectadas en el servidor.
- Se logró realizar la gráfica del comportamiento de incidentes para rechazos de autenticación en equipos inalámbricos con *LOKI* + *Grafana* empleando *Logql*.

Sobre los límites en la correlación de información para su presentación gráfica, se identificó que la información recolectada se puede presentar aplicando métodos de:

- Conteo de mensajes.
- Graficación numérica.
- Suma por periodo de tiempo.
- Filtrado para enfocar resultados.
- Multiplicación de indicadores para resaltar información.

Se observan limitantes en la cantidad de información contabilizada por periodos de tiempo. Como se muestra en la figura 4, se comprobó la posibilidad de generar conteos de ocurrencias en eventos registrados, pero se observó que entre mayor es la cantidad existente, se requiere de más procesamiento para generar los gráficos, por lo que es necesario considerarlo al plantear el monitoreo de grandes volúmenes como puede ser el consultar sobre un servicio del que se requiere graficar un tipo de acceso en particular. Es el caso de la figura 4, que muestra el análisis de datos de eventos de red, con un evento donde se eleva el tiempo de respuesta hacia *Google* y la posible relación con el incremento para un servicio de traducción de nombres de dominio.

Figura 4

Grafana, Gráficas numéricas y de calor + Umbrales

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 180 - 184

En cuanto al filtrado de información, se detecta que el lenguaje *LogQL* tiene una semántica limitada para aplicar filtros que permitan graficar cualquier por medio de filtros eventos contabilizados; sin embargo, acepta la integración de expresiones regulares con la lentitud que acarrea su uso para la generación de gráficas.

Finalmente, como se observó en la figura 4, la representación de gráficas se acota a las existentes al interior del sistema *Grafana*, y aunque pueden personalizarse, no se encontró posibilidad para generar una versión completamente personalizada.

Sobre recomendaciones a futuro y siguientes pasos, se identificó que algunos plugins para *Grafana* solamente están disponibles para la versión comercial. Asimismo, para el almacenamiento se recomienda buscar explotar de una mejor forma la base *FluentD* ya que permite la integración de información de más fuentes de datos (aplicaciones, servidores e información de monitoreo), pero se sugiere considerar que su configuración es más compleja.

Es importante hacer una evaluación de las limitaciones de los tableros de control, ya que hasta el momento se identifica la necesidad de buscar superponer información de diferentes fuentes de datos en una misma gráfica y en más de tres gráficas, para ampliar la visibilidad de la información en un solo tablero.

Sobre la generación de reportes automatizados de la información, se sugiere considerar que actualmente se puede compartir la información en tiempo real, pero para un reporte ejecutivo podría ser importante generar un reporte con formato con una descripción de los elementos graficados.

6. CONCLUSIONES

A partir de las pruebas realizadas y la conformación de la arquitectura para observabilidad de los eventos de red, se concluye que la arquitectura evaluada permite concentrar y presentar información operativa relevante y en tiempo real para la solución de incidentes.

La elección del *software* para la presentación gráfica de información recolectada tiene como base el sistema *Grafana* que se reconoce por la facilidad de crear tableros de control a partir de diferentes fuentes de información, apoyando la identificación de posibles incidentes de red además de permitir el establecimiento de umbrales gráficos, permitiendo identificar su relación con otros eventos y ayudando a visualizar el impacto de los eventos graficados a primera vista.

Se confirma también que con apoyo de las interfaces de *software* es posible relacionar distintos tipos de fuentes de información generando paneles gráficos sobre sucesos que ocurren en tiempo real, gracias a que los eventos se reciben y almacenan en línea en bases de datos que emplean marcas de tiempo para identificar su información.

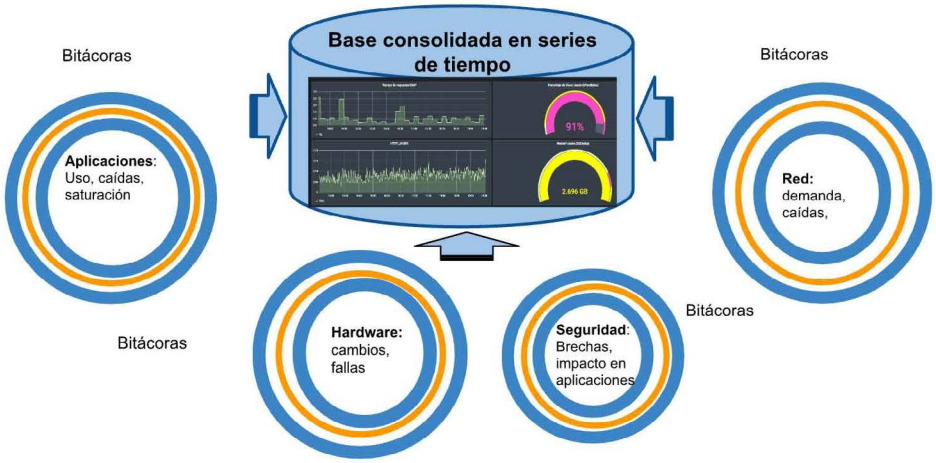
Se concluye también que la personalización de distintos paneles contribuye a la agilidad de acceso a la información tanto al personal operativo como directivos, debido a que las gráficas pueden adecuarse para presentarse de acuerdo con las necesidades de los usuarios con apoyo de filtros y aplicación de formatos de presentación de *Grafana* para destacar información relevante.

A partir de la cantidad de información que la arquitectura puede concentrar, se sugiere plantear un proyecto donde distintas áreas puedan concentrar información operativa creando un indicador de disponibilidad global no solo con la intención de identificar incidentes sino generando métricas de uso y saturación conjuntas como se representa en la figura 5:

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 181 - 184

Figura 5

Fuentes de datos que podrían integrarse para evaluar comportamientos



El trabajo conjunto puede facilitar también la identificación de las causas que originan un incidente al contar con información de los servicios a diferentes niveles, apoyando así la mejora de la disponibilidad de los servicios en red de la UNAM y el acceso a Internet.

REFERENCIAS BIBLIOGRÁFICAS

Chuck Lane and Joerg Linge, *PNP4Nagios*. Recuperado 30 de agosto de 2023 de https://github.com/pnp4nagios

Cisco Systems, Inc., (s.f.). *¿Qué es la observabilidad?* Recuperado 30 de agosto de 2023, de https://www.cisco.com/c/es_mx/solutions/full-stack-observability/what-is-observability.html

Cloud Arquitecture Center, GoogleCoud, Medición de DevOps: supervisión y observabilidad, (2023). *Cloud Architecture Center*, Última actualización: 2023-08-19 (UTC). Recuperado 30 de agosto de 2023 de https://cloud.google.com/architecture/devops/devops-measurement- monitoring-and-observability?hl=es-419

Grafana Labs (2023). *Grafana Loki documentation*. Recuperado 30 de agosto de 2023, de https://grafana.com/docs/loki/latest/

Grafana Labs (2023). *Technical documentation for Grafana Labs products and services*. Recuperado 30 de agosto de 2023, de https://grafana.com/docs/

FluentD Project team, Guides and Receips, Log Analytics. Recuperado 30 de agosto de 2023, de https://www.fluentd.org/guides

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 182 - 184

Marko Hartikainen. (2023). Bachelor thesis Degree *Programme in Business Information Technology* 2023, Häme University of Applied Sciences. Recuperado de https://www.theseus.fi/bitstream/handle/10024/804045/Hartikainen_Marko.pdf?sequence=2

Rainer Gerhards (GroBrinderfeld), rsyslog. *The rocket-fast system for log processing*. Recuperado 30 de agosto de 2023, de https://www.rsyslog.com/doc/master/index.html

ANEXO A

Tabla 1

Evaluación MoScoW. Evaluación de las características para selección de software

Característica disponible	Evaluación MoSCoW	Importancia
Soporte de almacenamiento por medio de base de datos libre y de alta velocidad de lectura y escritura	Debe tener la capacidad (Must)	1. La base de datos creciente de información. 2. Acceso múltiple para crear reportes personalizados. 3. Escritura constante 24x7x365
Licencia libre y posibilidad de escalar a servicio de paga	Debe tener la capacidad (Must)	Debido a que la Universidad maneja tráfico de usuario final superior a 20 Gigabits por segundo distribuidos adicional a los enlaces al interior de la república se generan datos operativos en su mayoría de poca relevancia operativa, no es viable pagar por cada evento registrado
Interfaz o mecanismos de conexión con fuentes de información externas.	Debe tener la capacidad (Must)	Es necesario que la información operativa de eventos relacionados con incidentes de red pueda vincularse con información de diferentes fuentes para generar reportes útiles.
Tableros de control web con facilidad de uso para los usuarios expertos y no expertos	Debe tener la capacidad (Must)	La presentación de los informes es tan importante como la generación de información debido a la importancia de contar con ellos al momento de identificar un comportamiento anómalo
Cambio de formato de información recolectada	Puede tener, pero no necesaria (Should)	Para presentar resultados es importante contar con un formato que refleje los hallazgos y se pueda identificar fácilmente a la vista del personal.
Alertas vía correo	Puede tener, pero no necesaria (Should)	El envío de información relevante por medio de correo electrónico es de gran ayuda cuando se encuentra el personal fuera del horario laboral.

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 183 - 184

Característica disponible	Evaluación MoSCoW	Importancia
Alertas por cualquier medio	Debe tener la capacidad (Must)	El envío de información relevante por medio de correo electrónico es de gran ayuda cuando se encuentra el personal fuera del horario laboral.
Módulos para ubicar geográficamente los dispositivos	Debe tener la capacidad (Must)	Esta característica es útil para ubicar si un incidente afecta geográficamente a otros sin una relación directa
Cálculos sobre información generar nueva información	Debe tener, pero no necesaria (Should)	Esta función ayuda a resaltar dentro del contexto de un reporte gráfico los elementos más importantes al presentarse un incidente de red.
Reportes personalizados	Debe tener, pero no necesaria (Should)	Importante para cuando se requiere generar un reporte ejecutivo a personal directivo y a usuarios finales.
Base de datos basada en series de tiempo circular	No debería tener (Won´t)	Es una característica deseable para datos que crecen muy rápidamente, esta característica sin embargo puede sustituirse por depuraciones periódicas.
Monitoreo en tiempo real	Debería tener (Could)	Entre más rápido se tenga la información de identificación de un incidente es mejor. Sin embargo, es aceptable el retraso hasta de un minuto dependiendo la cantidad de información analizada.
Tablero de control mostrando resultados en tiempo real	Debería tener (Could)	Entre más rápido se tenga la información de identificación de un incidente es mejor. Sin embargo, es aceptable el retraso hasta de un minuto dependiendo la cantidad de información analizada.
Operaciones con información obtenida para generación de gráficas.	Debería tener (Should)	Importante para cuando se requiere generar un reporte específico, ejecutivo a personal directivo y usuarios finales.
Alta disponibilidad	Puede tener, pero no necesaria (Could)	Es necesario que la información siempre esté disponible y es común que el sistema de monitoreo se vea afectado ante un incidente de red; es de gran utilidad contar con un respaldo activo del sistema.
Alertas gráficas al alcanzar umbrales	Puede tener, pero no necesaria (Could)	La preparación de un escenario en que debiera ser notificado el personal cuando se alcance el consumo de algún servicio presenta gran utilidad al momento del diagnóstico operativo de un incidente de red.

https://doi.org/10.22201/dgtic.ctud.2023.1.1.17 Vol. 1, núm. 1. Octubre-diciembre 2023, págs. 184 - 184

Tabla 2

Selección de software. Resultado de la evaluación de software para observabilidad

Software	Función principal	Resultado MoSCoW	Observaciones
Fluentbit	Almacenar información	NO INCLUIR	Complejidad alta y bajas posibilidades de integración
FluentD	Almacenar información	INCLUIR	Cumple con necesidades
NXlog	Recolectar Syslog	NO INCLUIR	Complejidad alta
Graylog	Recolectar Syslog	NO INCLUIR	Sistema parcialmente propietario
Octopussy	Recolectar Syslog	NO INCLUIR	No se adapta a otras soluciones
Telegraf	Recolecta información de sistema	INCLUIR	Se puede adaptar a otras soluciones
Grafana	Genera tableros de control	INCLUIR	Cumple con necesidades
Kibana	Genera tableros de control	NO INCLUIR	Sistema parcialmente propietario
Rsyslog	Recolectar Syslog	INCLUIR	Cumple con necesidades
Loki	Recolectar Syslog	INCLUIR	Cumple con necesidades
PNP Nagios	Recolecta información del sistema de monitoreo Nagios	INCLUIR	Cumple con necesidades