TP ESPECIAL

Objetivo

El objetivo del TP Especial es que cada grupo implemente un sistema de red específico y muestre su funcionamiento frente al resto de los alumnos y docentes junto con una exposición oral del tema tratado.

Temas

Cada grupo deberá implementar solo uno de los siguientes TP Especiales los cuales serán sorteados. Existe la posibilidad de presentar una idea nueva de tema a la cátedra de la materia que deberá ser aprobada para su realización.

Puntos a desarrollar

Cada grupo deberá implementar, demostrar y explicar, sin excepción, cada uno de los puntos que se especifican a continuación dentro del tema que haya elegido o le haya sido designado por la cátedra cómo mínimo para la aprobación del TP.

Tema 1 - Streaming

a) Configurar un servidor de streaming de Audio y Video con posibilidad de grabación de la sesión.
b) Pruebas con al menos dos fuentes simultáneas en vivo y diferidas.
c) Informar cantidad de clientes conectados y consumo de ancho de banda individual y total.
d) Mostrar el flujo de datos entre servidor y clientes, para dos o más usuarios conectados.
e) Explicación y demostración de codecs utilizados y utilización de ancho de banda de cada uno.
f) Mostrar protocolos utilizados (ej: RTP, RTSP, RTMP)

Tema 2 – WAF (Web Aplication Firewall)

a) Configurar un servidor Proxy que funcione como proxy reverso para recibir las peticiones para al menos 2 servidores con web server.
b) Configurar un servidor con ModSecurity que reciba las redirecciones del Proxy y chequee la seguridad de las mismas
c) Configurar al menos 3 reglas de solo detección para realizar análisis.
d) Configurar al menos 3 reglas de bloqueo.
e) Probar al menos 3 ataques para mostrar la respuesta del waf, configurar un página default de respuesta ante detección de anomalía.

TP ESPECIAL

Tema 3 – PeerTube

a) Montar al menos dos servidores de streaming
b) Mostrar sincronización entre servidores.
c) Mostrar cómo al menos 3 clientes comparten data al conectarse al menos a dos servidores y realizar P2P.
d) Mostrar tráfico con Wireshark de protocolos que intervienen.

Tema 4 - OpenVas/Greenbone

a) Instalación y Configuración.
b) Mostrar scaneo remoto con o sin login.
c) Mostrar el indicador de cambios en base a los últimos escaneos.
d) Mostrar detección de parches no aplicados y aplicar solución.
e) Mostrar detección de servicios en puerto no conocidos (ej SSH en port 2222)
f) Mostrar escaneos previos para no demorar la presentación.

Tema 5 – Zabbix ó Grafana (Sistema de monitoreo y alertas)

a) Crear una red virtual o real a monitorear, de al menos 4 hosts.
b) Configurar el monitoreo de al menos dos características de hardware de un servidor.
c) Configurar el monitoreo de un servicio web, usando pasos secuenciales dentro de la navegación de sitio.
d) Configurar alertas por tiempos de respuestas en un sitio web.
e) Configurar monitoreo de al menos 2 servicios (API REST, HTTP, SMTP,etc)
f) Configurar alarmas con distintos niveles según el tiempo de caída de servicio a distintos administradores u operadores.

Tema 6 – OPNSense ó pfSense (Router y firewall)

a) Se deben crear dos escenarios, uno en modo transparente (bridge) y otro en modo gateway.
b) Crear y aplicar políticas de QoS de al menos 3 servicios(ej. http, ftp, p2p)
c) Crear y aplicar políticas por aplicación, zoom, utorrent, etc.
d) Crear y aplicar políticas por tipo de tráfico, streaming, chat.
e) Configurar ancho de banda máximo y garantizado. Saturación del enlace para las pruebas.

TP ESPECIAL

Tema 7: OpenVPN

- Se deben crear tres tipos de VPN (Cliente-Sitio, Sitio-a-Sitio, Multisitio)
- Para la conexión Cliente-Sitio
    - El cliente debe pasar a través de un equipo que realiza NAT.
    - El cliente debe obtener una IP por DHCP de la red interna del sitio.
    - Probar la conexión directa por NAT y a través de un web Proxy
- Para la conexión Sitio-Sitio
    - Ambos equipos deben conectarse mediante direcciones “públicas” e interconectar sus redes internas. Los host de ambas redes deben tener distancia de un salto entre ellos.
    - Para la conexión multisitio debe interconectar al menos 3 sitios por VPN, los cuales deberán cumplir la topología "full mesh", es decir, que desde cualquier equipo de la red se puede llegar a cualquier otro equipo.

Tema 8: ELK

- Implementar ElasticSearch, Logstash y Kibana en un servidor para generar alertas y métricas al recibir información de servidores.
- Usar como pruebas al menos 4 tipos de servidores diferentes que contengan: Servidor Linux, Servidor Windows, servidor web, servidor de base de datos.
- Mostrar las ventajas y desventajas de la utilización o no de agente en los servidores que envían información.
- Recolectar información de tipo “Events” de Windows, syslog de Linux con y sin agente.
- Mostrar al menos las siguientes alertas:
    - Un usuario hace login desde una dirección IP no habitual.
    - Un servicio se encuentra caído o no responde hace x segundos.
    - Evento de firewall generado por un ataque.
- Mostrar al menos 3 reportes de kibana con estadísticas de una semana de actividad.
- Explorar alternativas open source a Logstash, por ejemplo Fluentd o Fluentbit. Muestre cómo se configuraría, similitudes y diferencias, pros y contras.

Tema 9: Serverless

- Diseñar una API Rest (serverless) con 3 endpoints, donde al menos debe haber 1 request POST, un request GET y un request GET del formato /name/[nombre-arbitrario]
- Los requests deben manejar e interpretar todo tipo de errores.
- Configure funciones (AWS Lambda, Azure Functions, GCP Functions) que atiendan a las solicitudes de la API y procese los datos de entrada.
- Todos los datos que se reciban en el body del POST deberán ser almacenados en una base de datos serverless.
- El body del POST deberá tener un valor “name”. Caso contrario debe devolver el error.

TP ESPECIAL

- Cuando se hace GET /name/[nombre-arbitrario], deberá traer al menos un campo de la base de datos (dato que se almacenó allí por haber hecho previamente un POST).
- Mostrar gráficos con métricas en CloudWatch (o similar) de todos los componentes serverless.
- Explicar cómo escala el sistema anterior.
- Explicar cómo funciona y las ventajas de usar un servicio Serverless frente a un servidor común.
- Explique cómo utilizaría el servicio de Lambda (o similar) en otros dos escenarios que no sea con el fin de una API.
- ¿Cómo se podría configurar un proveedor de identidad como Cognito (o similar) para atender requests autenticados mediante token JWT? ¿Que debe cambiar o que debe tener en cuenta a la hora de implementar el cambio?

Tema 10: Wireguard

- Se deben crear tres tipos de VPN (Cliente-Sitio, Sitio-a-Sitio, Multisitio)
- Para la conexión Cliente-Sitio
    - El cliente debe pasar a través de un equipo que realiza NAT.
    - El cliente debe obtener una IP por DHCP de la red interna del sitio.
    - Probar la conexión directa por NAT y a través de un web Proxy
- Para la conexión Sitio-Sitio
    - Ambos equipos deben conectarse mediante direcciones “públicas” e interconectar sus redes internas. Los host de ambas redes deben tener distancia de un salto entre ellos.
    - Para la conexión multisitio debe interconectar al menos 3 sitios por VPN, los cuales deberán cumplir la topología "full mesh", es decir, que desde cualquier equipo de la red se puede llegar a cualquier otro equipo.

Tema 11: Optimizador WAN (SD-WAN)

- Instalación de al menos tres aceleradores WAN a través de Internet con un producto a elección.
- Muestra de mejoras de tasas de transferencias sin límite de ancho de banda.
- Aceleración de al menos 3 protocolos diferentes (HTTP, CIFS, NFS, etc)
- Configuración de límite de ancho de banda. (Policy Shaper)
- Muestra de índices de deduplicación para cada una de las transmisiones realizadas.

Tema 12: Kubernetes

- Crear un cluster de Kubernetes de un Master y al menos dos slave, que exponga una API en un puerto genérico (distinto a 80)
- Implementar una base de datos local en un servidor y exponer un servicio que redireccione el tráfico del cluster al servidor.

TP ESPECIAL

- Deployar un web server (nginx o Apache HTTPD escuchando en el 80) y hacer un proxy reverso a la API.
- Mostrar dos versiones de API distintas conviviendo.
- Opcional: Integrar los servicios de Istio y Kiali al cluster.

Tema 13: Infraestructure as a Code (Terraform o similar)

- Mostrar cómo se configura, elementos necesarios para el despliegue y cómo configurar alguna de los siguientes escenarios:
    - Dos cuentas Cloud de distinto vendor (por ejemplo, AWS-Azure o AWS-GCP)
    - Una cuenta de AWS (o cualquier otro vendor) y un ambiente on-premise
- Crear un script/ejecutable que despliegue una red para el caso del ambiente Cloud
- Crear un script/ejecutable que despliegue un endpoint de un load balancer y un servicio por detrás (una API o un sitio web HTML sencillo).
- Muestre cómo se puede crear un ambiente activo-pasivo y como hacer el rollback a otra cuenta/sitio
- Muestre como deployar un sitio web estático con S3, CloudFront (servicios similares en otro vendor o análogo pero on-premise) y algún dominio propio. Para este último punto, puede optar por utilizar un dominio gratuito.
- Explique y muestre estrategias para manejar la seguridad de los componentes (por ejemplo, cómo se administran claves de base de datos o claves secretas de las cuentas de AWS)

Tema 14: CI/CD (Jenkins, CodePipeline o similar)

Recomendación: para este tema, puede trabajar en conjunto con el equipo de Serverless o Kubernetes. Lo puede llegar a necesitar.

Crear un pipeline que contenga los siguientes elementos:

- Tener un repositorio de código en Git que, cada vez que ocurre un push, buildear el código y deployar en un contenedor (on-premise con Docker, desplegarlo en Kubernetes on-premise o usando EKS o ECS en AWS), en una máquina virtual o en un servicio de ejecución de código sin servidor (por ejemplo, Lambda)
- Previo al deploy, ejecutar las pruebas unitarias del código.
- Enviar una notificación al equipo de DevOps si el despliegue fue correcto. Si fue incorrecto, enviar una notificación, detener el despliegue y hacer un rollback.
- Previo al despliegue en el ambiente, crear un proceso de aprobación manual a un grupo selecto de personas
- Muestre cómo hacer un despliegue en el mismo pipeline pero multi-ambiente. Por ejemplo, primero despliega el código en desarrollo y luego en producción.
- Extra: Explique y muestre que componentes podría utilizar para hacer el despliegue de una página web o app mobile y hacer un test de UX/UI

TP ESPECIAL

Consideraciones especiales

- El tipo de diseño y la forma de implementación serán discutidos entre el grupo y la cátedra durante las clases de laboratorio o teóricas, dejando la posibilidad de modificar este enunciado escrito, previo acuerdo entre el docente y los integrantes del grupo.
- Para la evaluación se tendrá en cuenta no sólo la implementación sino también la exposición oral y el documento para repetir la implementación (how-to)
- Todos los integrantes del grupo deben estar presentes en la presentación y ser oradores.
- Cualquier aclaración oral a cargo de la cátedra con respecto al enunciado del TP tiene la misma validez que el enunciado escrito.

Material a entregar

Cada grupo deberá subir el siguiente material al Campus en la sección respectiva de su grupo:

- Presentación PPT que se utilizará en la exposición
- Documento explicativo (how-to) de cómo se realiza la implementación.

Fecha de entrega, demostración y exposición oral

- El plazo máximo de entrega del TP es el **7 de junio a las 23:59 hs** vía Campus ITBA.
- Las presentaciones de los grupos se realizarán los días 8,13,15 y 22 de Junio en los horarios de la materia y según orden aleatorio obtenido mediante sorteo.
- Todos los integrantes del grupo deberán estar presentes en la exposición oral. No se tomarán exposiciones a grupos que no estén presentes todos sus integrantes y considerará desaprobado el TPE en la primera instancia. Siendo la próxima instancia el recuperatorio de TPE.