Unidad de desarrollo:

• Kafka es accedido desde los frontales y desde Storm a través del puerto 6667.

• La base de datos MySQL del entorno de producción es accedida por el puerto 3306 desde el frontal y desde Storm.

• Desde internet se accede al balanceador de carga y a los frontales del entorno de producción a través de los puertos 80 y 443.

• Desde el servidor VPN se debe acceder a todas las máquinas por el puerto 22.

• El servidor VPN debe estar accesible desde Internet a través del puerto 443.

• Las máquinas de Git, Redmine y Jenkins deben ser accesibles por los puertos 443 desde la red interna y desde la VPN para permitir trabajar a los desarrolladores.

• Las máquinas de la forja (Git, Redmine y Jenkins) deben poder acceder a la base de datos de la forja.

• La máquina de Jenkins debe poder acceder a todas las máquinas de producción a través del puerto 22 para permitir el despliegue automático.

Propuestas de medidas de seguridad:

IDS – SNORT: monitorización del tráfico de red entrante desde las redes externas. Tanto al servidor VPN, como al balanceador de carga, para prevenir ataques conocidos y sobre todo accesos no deseados.

Firewall: uso de firewalls para bloquear el tráfico no deseado. Desde la red interna a la forja de desarrollo, de la forja de desarrollo al entorno de producción e incluso dentro del entorno de producción. Tenemos ya definidos en la arquitectura los puertos que se pueden usar y en que direcciones y solo vamos a permitir ese tráfico.

IPS – Palo Alto: monitorizando el tráfico en busqueda de intrusiones no deseadas, tras el servidor VPN y el balanceador de carga (si hay firewall también tras él) tendremos un IPS como Palo Alto.

VulsRepo: para monitorizar nuestro sistema en busqueda de vulnerabilidades, usaremos vuls y vulsrepo. Como tenemos el sistema bastante automatizado, también es recomendable monitorizarlo en busqueda de vulnerabilidades. Para en caso de tenerlas, poder solventarlas lo antes posible.

Nagios: Las métricas son claves para la toma de decisiones. Vamos a tener monitorizando el sistema a nagios, visualizando los mismos en dashboard de fácil lectura en caso de que ocurran problemas criticas. Responder ante problemas con información siempre es clave.

Otras propuestas:

Otras propuestas que no tienen tanto que ver con la arquitectura pero que también me parecen interesantes son:

Permisos: Los permisos de las personas que acceden al sistema tienen que estar bien definidos. Puede parecer muy obvio pero merece la pena pensar bien los mismos que no lamentarnos en un futuro.

Formación: Formación en ciberseguridad para las personas de la organización. Según el perfil se puede proporcionar informacion muy técnica, como este master o charlas de buenas practicas a toda la plantilla.

TDD: En un sistema altamente automatizado, con integración y despliegue continuo, una muy buena practica de programación es usar Test Driven Development. Ya que antes de integrar o desplegar cualquier codigo deberemos pasar los test implementados para poder hacerlo. Si en estos test se tienen en cuenta temas de seguridad, vamos a poder prevenir vulnerabilidades y errores.