Trabajo Práctico N°3

Cifrado con clave publica/privada

1. Cliente de escritorio para correo electrónico, con plugin para PGP (Pretty Good Privacy).
   a. El correo debe ser leído unicamente por el destinatario.

      Con un juego de clave publica/privada generado por Enigmail, se probó el cifrado con clave publica del destinatario en clase. Funcionó como era esperado.

   b. El destinatario debe estar seguro que el correo lo envió el remitente.

      Con un juego de clave publica/privada generado por Enigmail, se probó la firma con clave privada del remitente. Funcionó como era esperado.

   c. Si no se usa encriptado, la tarea de desensamblar el email para convertirlo en uno para cada destinatario lo puede realizar el servidor de correos. Cuando se usa encriptado, este trabajo debe hacerlo el cliente de correos, ya que es el propietario de sus claves publica/privada, y publica de los destinatarios.

   d. ¿Que desafíos existen para implementar esta tecnología en clientes webmail?

      En estos casos, los algoritmos de cifrado debe trabajar del lado del servidor, como asi también usarlo de repositorio de claves. En este sentido, es necesario considerar si confiamos en nuestro servidor de correos para cuidar esta información.

      https://documentation.online.net/en/web/web-hosting/email-management/pgp-webmail

      La otra alternativa es escribir el email fuera del cliente web, con un plugin para navegador web que gestione claves y encripte, para luego transferir el resultado al cliente web.

      https://documentation.online.net/en/web/web-hosting/email-management/pgp-webmail

2. Autenticación de una computadora A usuario 1, a una computadora B usuario 2.

   Se realizo una experiencia con el gestor de contenedores docker (https://www.docker.com/). Este hypervisor se montó sobre windows 10 x64.

   a. Arquitectura

Para el despliegue del ejercicio se utilizaron dos imágenes ubuntu 16.04 (https://hub.docker.com/_/ubuntu/). Una servidor ssh y la otra cliente ssh.

De la imagen ubuntu se generó una imagen de transición con la utileria necesaria para el ejercicio (openssh, iputils, etc). De esta imagen se generan el cliente y servidor ssh.

b. Interacción entre hosts

- En el servidor se crearon todas las cuentas de invitados, aunque que solo se prepara con datos y se puede probar la cuenta hdonato (el cliente ssh)
- En el cliente se da de alta solo una cuenta de test (hdonato), se crea una clave asimétrica y se transfiere al servidor vía el comando scp (secure copy).
- La conexión al servidor requiere contraseña (de mismo texto que el nombre).
- Cuando el servidor recibe la clave publica de hdonato en /home/hdonato/.ssh/hdonato_id_rsa.pub, la agrega a su lista de confianza en authorized_keys con cat hdonato_id_rsa.pub >> authorized_keys.
- Las siguientes entradas al servidor ssh vía este protocolo, no requerirán contraseña para hdonato.

c. Referencias adicionales

Ademas del documento de apoyo dado en la practica, se consultaron:

- https://www.digitalocean.com/community/tutorials/how-to-configure-ssh-key-based-authentication-on-a-linux-server
- https://docs.docker.com/engine/examples/running_ssh_service/
- https://gist.github.com/dwilkie/f8d6526edc5f1a8aca385df5113567e4
- https://support.rackspace.com/how-to/checking-running-services-on-linux/
- https://www.ssh.com/ssh/sshd/
- https://stackoverflow.com/questions/27701930/add-user-to-docker-container?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

3. DKIM

DKIM es el resultado de la fusión de DomainKeys (desarrollador por Yahoo) e Identified Internet Mail (desarrollado por Cisco).

Se trata de un mecanismo para que un destinatario pueda validar los emails recibidos, comprobando de forma inequívoca que su origen es realmente el que aparece en las cabeceras del email.

Si bien es una tecnología de autenticación, sirve indirectamente para combatir el SPAM porque verifica la legitimidad del email.

DKIM incluye una cabecera en el mensaje con una firma digital del contenido del mensaje. Cuando se manda un email, el destinatario consulta al DNS del remitente y obtiene la clave pública del dominio. Con ella descifra la firma y obtiene un hash y la carga útil del mensaje.

Calcula el hash y lo compara con el recibido. Estos valores deben coincidir para comprobar autenticidad e integridad de extremo a extremo.

Como desventajas, se debe mencionar el overhead del los servidores (cifrar/descifrar en el emisor y el receptor).

Fuentes:

https://blog.mailrelay.com/es/2012/09/27/que-es-y-como-funciona-el-sistema-dkim

https://es.wikipedia.org/wiki/DomainKeys_Identified_Mail

http://www.dkim.org/