Proyecto final

Los proyectos están divididos en dos tipos:

1. Proyectos que consisten en el **desarrollo de un programa** que aplique criptografía para una función específica, empleando para ello la API criptográfica de Java (6 opciones disponibles).
2. Proyectos en los cuales se tiene que hacer una **implementación/configuración** de alguna herramienta y analizar su desempeño.

1. Transferencia segura de archivos con esquema de clave simétrica. Deben desarrollarse dos programas, uno cliente y uno servidor. El programa servidor debe escuchar por un puerto determinado, y esperar la conexión del cliente. El cliente recibe un nombre de archivo como parámetro. Una vez conectados cliente y servidor, el cliente debe negociar una clave de cifrado con el servidor empleando el algoritmo Diffie-Hellman (convencional o de curvas elípticas), y luego transferir el archivo empleando el algoritmo AES con clave de 256 bits, usando la clave previamente negociada. Al final del proceso el cliente debe calcular el hash SHA-256 del archivo que acaba de transmitir, y enviarlo al servidor. El servidor debe calcular el hash sobre el archivo recibido, y compararlo con el hash recibido del cliente. Si son iguales, debe indicarse que el archivo se transfirió adecuadamente.

2. Transferencia de archivos con chequeo de integridad con clave pública. Deben desarrollarse dos programas, un cliente y un servidor. El programa servidor debe escuchar por un puerto determinado, y esperar la conexión del cliente. El cliente recibe un nombre de archivo como parámetro. Una vez conectados cliente y servidor, el servidor debe generar un par de claves RSA (pública y privada), y mandar la pública al cliente. El cliente debe entonces generar una clave aleatoria de 256 bits y enviarla cifrada al servidor con la clave pública recibida; acto seguido debe enviar al servidor el archivo cifrado, empleando el algoritmo AES con clave de 256 bits, usando la clave recién enviada. El servidor descifrará la clave de 256 bits empleando la clave privada RSA, y luego descifrará el archivo empleando la clave de 256 bits. Al final del proceso el cliente debe calcular el hash SHA-256 del archivo que acaba de transmitir, y enviarlo al servidor. El servidor debe calcular el hash sobre el archivo recibido, y compararlo con el hash recibido del cliente. Si son iguales, debe indicarse que el archivo se transfirió adecuadamente.

3. Chat cifrado. Deben correrse dos instancias del programa en computadores diferentes. Las dos instancias deben conectarse por la red; una vez lograda la conexión, deben negociar una clave de cifrado empleando el algoritmo Diffie-Hellman (tradicional o de curvas elípticas). Esta clave debe ajustarse a 256 bits, para emplearla con el algoritmo AES. Toda la conversación entre las dos instancias del programa de chat debe ir cifrada a partir de ese momento.

4. Firmador y verificador de firmas. El programa debe tener tres opciones: 1) Generación de par de claves RSA. Debe generar la clave pública y la privada en dos archivos separados. El archivo de la clave privada debe quedar protegido con una contraseña. 2) Firmar archivo. Esta opción recibe como entradas un archivo cualquiera, y el archivo de clave privada. Una

vez comprobada la contraseña de bloqueo de la clave privada, el programa debe generar la firma digital del archivo, y guardarla en un archivo aparte. 3) Verificación de firma. Esta opción debe recibir como entradas el archivo original, el archivo que contiene la firma y el archivo de clave pública. Con estas tres entradas, debe verificarse que la firma sea correcta.

5. Cifrador/descifrador de archivos. El programa debe tener dos opciones: 1) Cifrar archivo. Debe recibir como entrada un archivo cualquiera, y una contraseña. A partir de la contraseña, debe generarse una clave de 256 bits, empleando el algoritmo PBKDF2. Por último, el archivo debe cifrarse con el algoritmo AES, usando la clave obtenida; el resultado debe escribirse a otro archivo, que debe contener también el hash SHA-256 del archivo sin cifrar. 2) Descifrado: Debe recibir como entrada un archivo cifrado y la contraseña. El programa deberá descifrar el archivo y escribir el resultado en un archivo nuevo. Luego, debe computar el hash SHA-256 del archivo descifrado y compararlo con el hash almacenado con el archivo cifrado, para verificar la integridad del proceso.

6. Funcionalidad de login para una plataforma. Este programa debe permitir gestionar los nombres de usuario y contraseñas de una plataforma cualquiera. Debe tener dos tipos de usuarios: el administrador (debe haber solamente uno) y los usuarios comunes. El usuario administrador debe poder consultar los nombres de los usuarios existentes, eliminar un usuario o poner en blanco la contraseña de un usuario. Los usuarios comunes deben poder consultar su última fecha/hora de login, y cambiar su contraseña. Las contraseñas deben almacenarse en un archivo de texto o base de datos, empleando salt. Se sugiere investigar y emplear el algoritmo PBKDF2 para el hashing de las contraseñas.

7. Proyecto Configuración de un Honeypot

Para este proyecto el grupo debe documentar el proceso de configuración del honey pot y documentar/explicar claramente al menos cuatro vulnerabilidades que se dejan abiertas en el honey pot, y como se puede verificar que el hacker está trabajando en el honey pot.
En la sustentación tienen que explicar claramente como se explotan dichas vulnerabilidades.

8. Comparación de herramientas IDPS basados en host

Para este proyecto cada grupo debe tomar dos herramientas (Tripwire, Nagios , Ossec u otra que investigue) y realizar su proceso de instalación claramente documentado.
Ya cuando tenga las herramientas instaladas realizar una comparación entre las dos con respecto a:

- Facilidad de proceso de instalación
- Funcionalidades disponibles, en este caso revisar funcionalidades básicas.
- Que tan amigable es la herramienta
- Cuanto carga el procesamiento del servidor.
- Revisar si las herramientas tienen vulnerabilidades
- Funcionalidades adicionales que encuentre

Recomendaciones para la presentación del informe escrito y posterior sustentación:

Los estudiantes deben conformar equipos de 3 personas (o trabajar en los grupos que ya están conformados) para desarrollar este proyecto. No se permite ningún tipo de plagio. Si utiliza IA de especificar en que parte del trabajo la utilizó y el prompt(s) respectivo(s) que diseñó.

. Los entregables son:

- El código fuente del programa debidamente documentado.
- Un pequeño informe que incluya la manera cómo hicieron el programa/configuración, las dificultades que tuvieron, y conclusiones.

La fecha de entrega del proyecto es el **miércoles 4 de Junio, en horas de la mñana**. Se harán sustentaciones de **20 minutos por Zoom**; oportunamente les enviaré una encuesta para que escojan el horario de sustentación.