Ejercicio 1

El concejo deliberante del partido de Tecnociudad solicitó la construcción de un sistema distribuido que permita realizar referendos (votaciones de toda la población) asiduamente y transparentar algunos actos de gobierno.

El sistema requiere poder garantizar que no haya votos duplicados, y al mismo tiempo garantizar el anonimato de las votaciones.

El sistema propuesto es el siguiente:

- Cada persona elige un numero aleatorio, y genera un identificador unico IDp = H(Rp || D.N.I.) donde H(x) es una función de hash libre de colisiones.
- A la hora de votar el usuario carga el IDp y el sistema registra el voto con el par (IDp, valor votado)
- Al terminar la elección se cuentan los votos, y se publica el padrón de IDps asociado a cada resultado.
- Cualquier persona puede verificar que su voto se computo correctamente consultando el padrón, también se puede observar que no haya IDps duplicados

a) Explicar por qué nadie puede saber el voto de una persona (un DNI) en particular (1 pto.)

b) Pensar y explicar cómo es posible en este sistema que una persona efectúe múltiples votaciones (1 pto).

c) Proponer una mejora al sistema que evite el problema del punto anterior, sin perder la propiedad de anonimato (1 pto).

Ejercicio 2

Considerar el siguiente algoritmo que puede ser utilizado para que dos sistemas se puedan sincronizar en un valor, 0 o 1,sin que ninguno de los dos pueda influenciar el valor:

1. A genera r1 y envía a B H(r1)
2. B genera r2 y envía a A H(r2) cuando ambos reciben el valor de la otra parte
3. A envía r1 a B
4. B envía r2 a A

A y B calculan x = (r1 & 1) xor (r2 & 1) (o sea el xor entre los bits menos significativos de r1 y r2)

a) Explicar por qué son necesarios los pasos 1 y 2. ¿Qué verificación adicional deberían hacer A y B no mencionada en la descripción anterior? (1 pto.)

b) Suponer que B reciba H(r1) y reenvía ese valor a A en el paso 2 ¿que conseguiría hacer? Proponer alguna modificación que lo evite (1,5 ptos.)

c) Modificar el algoritmo para que A y B se pongan de acuerdo en el valor de un byte en lugar de un bit. (0.5 ptos).

Ejercicio 3

Agrolin S.A. Se dedica a la construcción de soluciones tecnológicas para el campo.

En estos momentos está desarrollando una nueva línea de productos que automatizan la cosecha. El producto insignia, robo-cosecha, consiste en un sistema compuesto por dos drones, una cosechadora autónoma y un sistema de control integrado.

El cliente delimita en el sistema de control la zona a cosechar indicando los límites en un mapa. Con esa información, un dron sobrevuela el terreno extrayendo imágenes que permiten determinar las zonas aún no cosechadas. con el mapa de zonas a cosechar se programa el recorrido de la cosechadora. Como medida adicional, el segundo drone sobrevuela el camino delante de la cosechadora para detectar movimientos o anomalías y de esta manera enviar la información para que la cosechadora frene y no ocurra ningún accidente.

Los sistemas de mapeo de campo, análisis de cosechas y conducción autónoma de la cosechadora, ya existen, dado que fueron desarrollados por separado anteriormente como productos independientes. Ya han sido probados en campo y funcionan correctamente.

El diferencial de este nuevo producto es integrar todo y conectarlo de forma que ocurra automáticamente todo el proceso. Para ello, la base donde corre el software se comunica punto a punto con los drones y la cosechadora y actúa como el coordinador de la operación. Las conexiones son por radiofrecuencia con un protocolo basado en datagramas que da una abstracción similar a UDP. Sobre eso se construyó una capa segura que brinda autenticación e integridad, similar a TLS (o mejor dicho a dTLS, la variante de TLS pensada para datagramas). El software de la base está desarrollado en Java. El de los drones y la cosechadora, en C++.

Siguiendo la metodología de hipótesis de falla, establecer 4 hipótesis de vulnerabilidades que tengan alta probabilidad de existir en el escenario descrito. Para cada una, describir únicamente la hipótesis y la prueba que realizan para confirmar o refutar (1 pto. por hipótesis).

Ejercicio 1

a. Esto se debe a que se hashea no solo el DNI sino que tambien un numero aleatorio que no se conoce. Entonces por mas que alguien haga el hash del DNI de otra persona no va saber a quien voto porque no existe un IDp = H(DNI).

b. Un usuario podria elegir multiples numeros aleatorios y generar multiples IDPs. Luego, a la hora de votar, el usuario carga los distintos IDPs y asi registra un voto por cada IDp. Dado que las funciones de hash no son inversibles nadie podria saber que hasheo su DNI con distintos numeros.

c. En vez de cada persona elija un numero aleatorio, el gobierno puede enviar un Rp a cada uno. Entonces, si algun usuario intenta votar multiples veces, el gobierno sabe todos los hashes existentes y puede descartar aquellos que no deberian estar.

Ejercicio 2

a. Los pasos 1 y 2 son necesarios para que cada uno pueda verificar que el otro no cambio su valor. Es decir, si A manda $H(r_1)$ y luego envia $r_1' \neq r_1$, B puede hacer $H(r_1')$ y verificar que $H(r_1') \neq H(r_1)$. Esto se debe a que cualquier cambio en la entrada provoca un cambio en la salida de la funcion de hash.

b. En este caso, B consigue que $x = 0$. Esto se debe a que cuando B reciba r_1 en el paso 3, se lo envia a A y luego $x = (r_1 \,\&\, 1) \text{ xor } (r_1 \,\&\, 1) = 0$.

c. Exactamente lo mismo solo que $x = (r_1 \,\&\, 11111111) \text{ xor } (r_2 \,\&\, 11111111)$.

Ejercicio 3

(1) Allocation of Resources without limits or Throttling

Amenaza: Denial of Service

Esta vulnerabilidad ocurre cuando el sistema no hace los chequeos necesarios en cuanto a los recursos solicitados por los usuarios. En este caso, un atacante podria solicitar un gran numero de conexiones hasta agotar los recursos y asi evitar que los drones puedan conectarse.

Pruebas: en un ambiente de prueba

Intentar conectarse multiples veces con el mismo usuario y ver si el sistema acepta todas las conexiones entrantes.

(2) Improper authentication

Amenazas: tampering, information disclosure y denial of service

Esto se da cuando el sistema no implementa correctamente las funciones de autenticacion. Un atacante puede generar conexiones si consigue impersonar un dron.

Pruebas:

- Intentar conectarse sin autenticacion
- Intentar conectarse con usuarios y contraseñas comunes

(3) Sensitive Data Exposure

Amenazas: information disclosure, tampering y spoofing

Como la comunicacion entre componentes usa TLS v1, un atacante puede obtener las credenciales de algun dron e impersonarlo.

Prueba:

Intentar desencriptar los mensajes capturados en la comunicacion e intentar obtener informacion sensible.

(4) Replay Attack (?)

Dado que los datagramas no tienen nada que los identifique, un atacante podria interceptar alguna señal de frenado y reenviarla cuando las cosechadoras esten en uso.

Prueba:

Capturar algun datagrama de señal de frenado, reenviarlo y ver si la cosechadora frena.