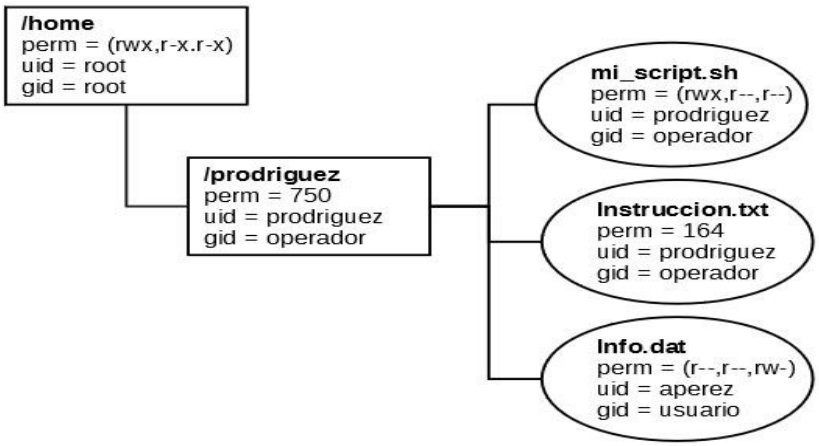
Fundamentos de la Seguridad Informática

Segundo Parcial

Sistemas Operativos

1. **Unix: Sujetos**
   a) Cómo se identifican los sujetos en UNIX
   b) Los sujetos en UNIX tienen asociado un UID real y otro efectivo. ¿Qué son y para qué sirven?
   c) Todos los sistemas UNIX cuentan con un programa para cambiar la contraseña del usuario (*passwd*). ¿Por qué este programa lleva el permiso Set User Id (SUID).
2. **Windows: Objetos**
   a) Describa qué información o estructuras de datos llevan los objetos en Windows para implementar las reglas de control de acceso.
   b) ¿Qué sucede si un sujeto desea acceder a un objeto que no tiene definido una DACL?
   c) Cuando vimos el algoritmo de control de acceso en Windows, se indicó que las reglas negativas prevalecen sobre las positivas. ¿Cómo vimos que se implementa esto?
3. **Unix: permisos**
   El usuario *prodriguez* tiene HomeDir */home/prodriguez*, gid primario *operador* y secundario *usuario*. Dada la siguiente información de la estructura de archivos y permisos.



   Responda cual será el resultado de las siguientes operaciones de control de acceso, justificando en cada caso su repuesta.
   a) Un proceso del usuario *prodriguez* desea modificar el archivo */home/prodriguez/instruccion.txt*
   b) Un proceso del usuario *prodriguez* desea modificar el archivo

/home/prodriguez/info.dat

c) Un proceso del usuario *ahernandez* que conoce la ruta del archivo */home/prodriguez/mi_script.sh* desea abrir el archivo para ver su contenido. Suponga que *ahernandez* no pertenece a los grupos *operador* o *usuario*
d) Si *prodriguez* quisiera que todos los archivos que crea con su usuario no contengan por defecto escritura para el grupo y ningún permiso para el resto del mundo, que mascara debería establecer.

Redes

4. ARP
   a) Describa brevemente el protocolo ARP
   b) ¿Que problema de seguridad presenta esté protocolo? Describa un posible ataque al mismo.
   c) ¿Que contra-medida vimos en el curso que se puede implementar para mitigar el ataque mencionado en (b)?
5. IPSec
   a) Este protocolo provee o implementa dos mecanismos (o protocolos) de seguridad. ¿Cuáles son estos mecanismos y en que se diferencian?
   b) El protocolo define la noción de asociaciones de seguridad (SA). Indique como están definidas y para que se utiliza.

6. Firewall
   a) ¿Qué es un firewall? Indique cuál es su función básica.
   b) Supongamos que estamos realizando la instalación y configuración de un firewall perimetral entre la red LAN y la conexión WAN (Internet) de una empresa. ¿Que situaciones o amenazas que vimos durante el curso nos pueden llevar a considerar el definir reglas de filtrado de conexiones salientes?

7. Intrusion Detection System (IDS)
   a) ¿Qué es un Sistema de Detección de Intrusos (IDS) ?
   b) ¿Qué tipos de IDS hay?. Explique.

8. Honeypots
   a) ¿Qué es un Honeypot?
   b) ¿Qué diferencia hay entre los Honeypot de bajo y alto nivel de interacción?

Aplicaciones

9. Software
   a) Mencione dos causas básicas que llevan a fallas en la seguridad del software.
   b) ¿Qué es lo que debería aplicarse para *atenuar* las fallas de seguridad causadas por el software? ¿En dónde lo aplicaría?

10. Web
    a) ¿Cuándo de produce un ataque de XSS? Describa un escenario posible, indicando víctima y atacante.
    b) Describa el ataque de CSRF. ¿Cómo se mitigaría un ataque de CSRF?

Laboratorio

11. Seguridad en sistemas

    a) Dada las siguientes *flags* y un conjunto de definiciones, indique que *flag* corresponde con cada definición.

    Flags:

    1) Required 2) Requisite 3) Sufficient 4) Optional

    Definiciones:

    a) Indica que es necesario que el módulo tenga éxito para que la pila también lo tenga. Si se produce un fallo, el control se devuelve inmediatamente a la aplicación.

    b) Su valor será tenido en cuenta sólo en caso de que no se haya llegado a ningún valor concreto de éxito o fracaso.

    c) El éxito en este módulo, si no se ha producido un fallo en los procesados anteriormente en la pila, determina el éxito de la pila.

    d) Indica que es necesario que el módulo no tenga éxito para que la pila lo tenga. Si se produce un fallo, no se notifica hasta que se procesa el resto de la pila.

    e) Ninguna de las anteriores

    b) Realice una comparación entre **John the ripper** e **hydra**, indicando las características más importantes de cada software.

12. Seguridad en Redes

El siguiente fragmento de política fue escrito utilizando la herramienta *Firewall Builder*, donde se cumple que: **servidor-interno ≠ fsi-fw para toda dirección IP.**

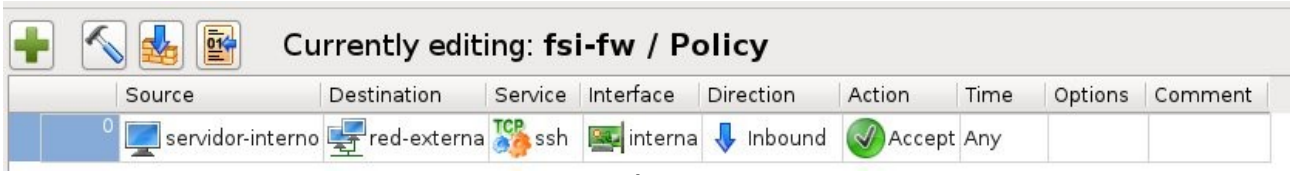


Imagen 1: Fragmento de Política escrita en Firewall Builder

Indique cuáles de las siguientes compilaciones a *iptables* son posibles:

1)

```
# ================ Table 'filter', rule set Policy
# Rule 0 (eth1)
#
$IPTABLES -A INPUT  -i eth1  -p tcp -m tcp  -s 10.0.0.2   -d
192.168.0.0/24   --dport 22  -m state --state NEW  -j ACCEPT

$IPTABLES -A FORWARD  -i eth1  -p tcp -m tcp  -s 10.0.0.2   -d
192.168.0.0/24   --dport 22  -m state --state NEW  -j ACCEPT
```

2)

```
# ================ Table 'filter', rule set Policy
#
# Rule 0 (eth1)
#
$IPTABLES -A FORWARD  -i eth1  -p tcp -m tcp  -s 10.0.0.2   -d
192.168.0.0/24   --dport 22  -m state --state NEW  -j ACCEPT
```

3)

```
# ================ Table 'filter', rule set Policy
# Rule 0 (eth1)
#
$IPTABLES -A OUTPUT  -o eth1  -p tcp -m tcp  -s 10.0.0.2   -d
192.168.0.0/24   --dport 22  -m state --state NEW  -j ACCEPT

$IPTABLES -A FORWARD  -i eth1  -p tcp -m tcp  -s 10.0.0.2   -d
192.168.0.0/24   --dport 22  -m state --state NEW  -j ACCEPT
```

NOTA: Justifique indicando las condiciones que se tienen que cumplir para los diferentes actores involucrados.

13. Seguridad en aplicaciones

a) Indique **al menos** dos vulnerabilidades del código debajo presentado, indicando los pasos para atacarlas.
b) ¿Cómo puede mitigar dichas vulnerabilidades?

```
1. #include <stdio.h>
2. #include <string.h>
3. #include <stdlib.h>
4.
5. #define BASE_DIR "/usr/share/fsi"
6.
7. int PasswordOk() {
8.           char GoodPassword = 'F';
9.           char Password[8];
10.
11.                    gets(Password);
12.                    if (!strcmp(Password, "GSIlab4"))
13.                              GoodPassword = 'T';
14.
15.                    return (GoodPassword == 'T');
16.          }
17.
18.          void PrintPasswd() {
19.                    FILE *in;
20.                    char c;
21.
22.                    in = fopen("/etc/passwd", "r");
23.                    if(in != NULL) {
24.                              while((c = fgetc(in)) != EOF) putchar(c);
25.

26.                              fclose(in);
27.                    } else
28.                              printf("No se pudo imprimir el archivo: /etc/passwd");
29.          }
30.
31.          void PrintFile(char* fileName) {
32.                    char fullName[256];
33.
34.                    strcpy (fullName, BASE_DIR);
35.                    strcat (fullName, "/");
36.                    strcat (fullName, fileName);
37.
38.                    printf("Archivo seleccionado: %s\n", fullName);
39.
40.                    FILE *in;
41.                    char c;
42.
43.                    in = fopen(fullName, "r");
44.                    if(in != NULL) {
45.                              while((c = fgetc(in)) != EOF) putchar(c);
46.

47.                              fclose(in);
48.                    } else
49.                              printf("No se pudo imprimir el archivo: %s\n",
```

```
  fullName);
50.          }
51.
52.          int main(int argc, char** argv) {
53.                    char User[8];
54.                    if (argc < 3) {
55.                              printf("Uso: %s user filename\n", argv[0]);
56.                              exit(0);
57.                    }
58.
59.                    strcpy(User, argv[1]);
60.                    printf("Ingrese la password: ");
61.                    if (PasswordOk()) {
62.                              printf("Bienvenido %s!!\n", User);
63.                              PrintFile(argv[2]);
64.                    } else
65.                              puts("Usuario y/o password incorrecta");
66.
67.              return 0;
68.          }
69.
```