

# esercizi su controllo di accesso e sicurezza di sistema

# access matrix e confidenzialità

- $S=\{a,b,c\}$
- $O=\{fa,fb,fc\}$
- $R=\{read,write\}$
  
- il modello è DAC o MAC?
- può **a** scrivere i file **fc**?
- può **c** leggere i file **fa**?
  
- può **c** venire a conoscenza del contenuto dei file **fa**?
  - come?

	<b>fa</b>	<b>fb</b>	<b>fc</b>
<b>a</b>	rw		
<b>b</b>	r	rw	
<b>c</b>		r	rw

# access matrix confidenzialità

- considera il problema della divulgazione delle informazioni di **fa** verso **fc** nelle seguenti matrici

	fa	fb	fc
a	rw		
b	r	rw	
c		r	rw

	fa	fb	fc
a	rw	w	
b		rw	
c		r	rw

	fa	fb	fc
a	rw	w	
b	r	rw	w
c			rw

	fa	fb	fc
a	rw	r	
b		rw	w
c			rw

	fa	fb	fc
a	rw	w	
b		rw	r
c			rw

	fa	fb	fc
a	rw	r	
b	r	rw	w
c		w	rw

# grafo del flusso “potenziale” di informazione

- per ciascuna delle precedenti matrici disegna il seguente grafo
  - nodi:  $f_a$ ,  $f_b$ ,  $f_c$
  - archi: da  $f_x$  a  $f_y$  (orientato) se esiste un soggetto che può leggere  $f_x$  e scrivere  $f_y$

# access matrix, $\text{grant}_x$ e confidenzialità

- $S=\{a,b,c\}$
- $O=\{fa,fb,fc\}$
- $R=\{\text{read},\text{write},\text{grant}_a, \text{grant}_b, \text{grant}_c\}$ 
  - $\text{grant}_x$  permette di cedere diritti a  $x$
- il modello è DAC o MAC?
- in questo stato
  - può **a** scrivere i file **fc**?
  - può **c** leggere i file **fa**?
- può **c** venire a conoscenza del contenuto dei file **fa**?
  - come?

	<b>fa</b>	<b>fb</b>	<b>fc</b>
<b>a</b>	$\text{rwg}_c$		
<b>b</b>		$\text{rw}$	
<b>c</b>			$\text{rw}$

# access matrix, $\text{grant}_x$ e confidenzialità

- considera il problema della divulgazione delle informazioni di **fa** verso **fc** nelle seguenti matrici

	fa	fb	fc
a	$\text{rwg}_b$		
b		rw	
c			$\text{rwg}_b$

	fa	fb	fc
a	$\text{rwg}_b$		
b		$\text{rwg}_c$	
c			rw

	fa	fb	fc
a	rw		
b		$\text{rwg}_{ac}$	
c			rw

	fa	fb	fc
a	$\text{rg}_b$		
b		rw	
c			$\text{rg}_b$

	fa	fb	fc
a	$\text{g}_b\text{w}$		
b		rw	
c			$\text{g}_b\text{w}$

# filesystem unix controllo di accesso

- considera i permessi dei seguenti file e directory e i processi attivi con le credenziali e le directory correnti indicate
- supponi che nessuno cambi i permessi
- P1 puo' ottenere la lista dei file in d2? e P2?
- quali file possono essere letti da P1 quali da P2?

<b>permessi</b>	<b>utente</b>	<b>path</b>
<b>d r-x</b>	<b>a</b>	<b>/</b>
<b>- r--</b>	<b>a</b>	<b>/file1</b>
<b>d r--</b>	<b>a</b>	<b>/d1/</b>
<b>d r-x</b>	<b>a</b>	<b>/d1/d2/</b>
<b>- r--</b>	<b>a</b>	<b>/d1/d2/file2</b>
<b>processo</b>	<b>utente</b>	<b>directory corrente</b>
<b>P1</b>	<b>a</b>	<b>/</b>
<b>P2</b>	<b>a</b>	<b>/d1/d2/</b>

# filesystem unix controllo di accesso

- considera i seguenti file e directory con i loro permessi
- supponi che i permessi siano immutabili
- dai la matrice di accesso  $S=\{P1, P2, P3\}$ ,  $O=\{file1, file2\}$ ,  $R=\{r,w\}$

permessi	utente	gruppo	path
d r-x r-x	a	g	/
- r-- rw-	a	g	/file1
d r-- r--	a	g	/d1/
d r-x r-x	a	g	/d1/d2/
- r-- rw-	a	g	/d1/d2/file2

Processo`	utente	gruppo	directory corrente
P1	a	g	/
P2	b	g	/
P3	b	g	/d1/d2



# suid

- per un utente diverso da pippo è possibile vedere i segreti di pippo?

permessi	utente	gruppo	path
<code>drwxrwxr-x</code>	<code>root</code>	<code>root</code>	<code>/</code>
<code>drwxrwxr-x</code>	<code>root</code>	<code>root</code>	<code>/home/</code>
<code>drwxrwxr-x</code>	<code>pippo</code>	<code>pippo</code>	<code>/home/pippo/</code>
<code>-rw-rw----</code>	<code>pippo</code>	<code>pippo</code>	<code>/home/pippo/segreti</code>
<code>-rwsrw-r-x</code>	<code>pippo</code>	<code>pippo</code>	<code>/home/pippo/cat</code>

- potrebbe un utente diverso da pippo cancellare o modificare i segreti di pippo?

# hard links

permessi	utente	gruppo	path
<code>-rw-rw----</code>	<code>pippo</code>	<code>pippo</code>	<code>/home/pippo/segreti</code>

- l'utente pippo da shell esegue

```
cd /home/pippo
```

```
ln segreti segreti2
```

- il comando `ls` che permessi mostrerà per il file `segreti2`?
- se pippo esegue...

```
chmod o+rw segreti2
```

- il comando `ls` che permessi mostrerà per il file `segreti`?

# bugs in login

- una implementazione di unix ha installato il comando “login” con un bug
- quando “login” legge il file /etc/passwd per identificare la shell da lanciare per l’utente si può verificare un buffer overflow
  - il buffer in cui si memorizza il nome della shell ha una lunghezza fissata
- che privilegi potrebbe riuscire ad ottenere un hacker?
  - ci sono almeno due risposte, in base alle scelte del progettista/programmatore di “login”

# sudo

- qual'è il significato delle seguenti configurazioni di sudo?

```
pizzonia ALL= ALL
```

```
pizzonia ALL= (pippo) ALL, (pluto) ALL
```

```
pizzonia ALL= NOPASSWD: /sbin/ifconfig eth0  
up, /sbin/ifconfig eth0 down
```

```
pizzonia ALL= NOPASSWD: /sbin/ifconfig, !  
/sbin/ifconfig eth0 *
```

```
pizzonia workstation = /usr/bin/*
```

```
pizzonia mega= /sbin/ifconfig ""
```

# syslog ed sms

- in una lan sono presenti i seguenti servizi su macchine distinte
  - www, smtp, imap
- si desidera avere tutti i messaggi di log su una quarta macchina
- si desidera ricevere i messaggi più critici (error, critical, alert, emergency) via sms
- per questo si è adibita una ulteriore macchina collegata ad un “trasmettitore di sms”
  
- mostra una architettura che sia in grado di fare ciò mediante l’uso di
  - syslog
  - un software in grado di eseguire una azione quando ad un file viene aggiunta una nuova linea (ad esempio “swatch”)