(۱)

الف) غلط؛ این مدل اشکالاتی دارد که کارایی‌اش را محدود می‌کند. گاهی اوقات یک subj. سطح بالا می‌خواهد به یک subj. سطح پایین، پیامی بفرستد و یک write انجام بدهد. ولی این کار در این روش ممکن نیست، و اصلاً practical نیست.

برای رفع این مشکل، اجازه می‌دهند که یک subj.، دو level داشته باشد؛ یعنی current و دیگری max، و هرگاه بخواهد به یک subj. در سطح پایین‌تر پیامی بفرستد، از max به current بیاید و این کار را بکند. ولی این کار هم ممکن است یک جریان اطلاعاتی رو به پایین را در این مدل ایجاد می‌کند، که این محرمانگی را نقض می‌کند. پس این حالت را به صورت optional برای این مدل قرار دادند. و یک tradeoff بین practical بودن و ضعف security ایجاد شد.

به طور کلی، مشکلات و محدودیت‌های BLP عبارتند از:

[1] فقط درباره‌ی confidentiality است. درباره‌ی integrity و availability اصلاً بحثی نکرده است، و برای سیستم واقعی و general قابل استفاده نیست.

[2] درست است که گفته می‌شود ترکیبی از DAC و MAC است، ولی اصلاً درباره‌ی DAC و جزئیات اجرای آن، چیزی نگفته است و نگفته این کنترل چگونه برقرار می‌شود.

[3] امکان ایجاد کانال پنهان وجود دارد، که با آن ارتباطات مسیری غیرمجاز برقرار می‌شود و محرمانگی نقض می‌شود.

البته به طور کلی برای استفاده از این مدل باید جزئیات بیشتری را دخیل کرد تا به صورت عملی قابل استفاده باشد و کارایی قابل قبولی فراهم کند.

(1)

ب) غلط: در این مدل مقصود داریم که وقتی ما به عنوان صاحب یک فایل، امکان دسترسی به آن را به یک Subj.1 دیگری نیز می‌دهیم؛ آن Subj.1 اجازه نداشته باشد از فایل مورد نظر ما یک کپی تهیه کند و خودش به اختیار خودش آن نسخه‌ی کپی شده را در اختیار Subj.2 دیگری بدهد.

اگر Subj.1 که ما دسترسی روی obj. خودمان را بهش داده‌ایم، یک کپی از obj. ما بگیرد، اگر Subj.2 بخواهد به این obj. دسترسی داشته باشد باید هم توسط ما و هم توسط Subj.1 این دسترسی به او تعلق بگیرد.

الف) در نتیجه این دسترسی که در این سوال توسط B روی فایل C به A تعلق گرفته است، کافی نیست. و باید A هم این دسترسی را به C بدهد برای تغییر در این فایل.

PACL این فایل برابر $PACL_A \cap PACL_B$

و PACL(C) باید برابر $PACL_{A,B}$ باشد. ولی فقط $PACL_B$ را دارد.

---

(1)

ج) غلط: دو مسئله‌ی صحت و محرمانگی کاملاً جدا از یک دیگر هستند. و نمی‌توان از روی یکی، دیگری را حدس زد یا رابطه‌ای بین این دو در نظر گرفت.

مدل Biba روی صحت اطلاعات متمرکز است. و مدلی را فراهم می‌کند که صحت اطلاعات را در سطوحی طبقه‌بندی می‌کند. در نتیجه هیچ حرفی درباره‌ی محرمانگی اطلاعات نمی‌زند و نمی‌توان اصلاً گفت که محرمانگی چگونه است در این مدل، و تضمینی در این راستا ندارد.

Day..... Month..... Year.....

(2)

الف) از روش capabilities استفاده می‌شود. در این مکانیزم، متناظر
با هر Subj، یک c-list وجود دارد که بیان می‌کند آن Subj.
چه مجوزهایی روی چه obj هایی دارد.

(2)

ب) در این حالت، چون procedure قصد دارد یک procedure از حلقه‌ای
متفاوت با سطح دسترسی متفاوت را اجرا کند، این اتفاق ممکن
است خطرناک باشد.
در این حالت، کنترل وارد عمل می‌شود و کنترل می‌کند که آیا آرگومان‌هایی
که برای فراخوانی استفاده می‌شود، درست است یا خیر؟! آیا شرط فراخوانی
درست است یا خیر؟! سپس، اگر همه چیز امن باشد، کنترل اجازه‌ی
کار را می‌دهد. در غیر این صورت، عبور این فراخوانی گرفته می‌شود.

(2)

ج) در این اصل بیان می‌شود که اگر پیش فرضی درست را روی سیستم داشته باشیم،
پیش فرضی امن و fail-safe باشد.
حال در مورد معرفی کردن permission هایی روی یک obj توسط یک Subj
در حال ACL؛ اگر در مجوزهای مربوط به یک obj، اسم یک Subj
موجود در سیستم را بیان نکرده باشیم، به صورت پیش فرض به این شکل
تعیین می‌شود که آن Subj، هیچ permission ای روی آن obj
ندارد. پس به این صورت، ما یک پیش فرض امن برای دسترسی‌ها در
نظر گرفته‌ایم. و فقط در صورتی مجوزی برای شخصی صادر می‌کنیم، که صراحتاً
گفته شده باشد آن شخص چه permission ای روی یک obj دارد.

3. کامند هایی که برای هر کدام از قسمت ها باید اجرا کنیم، بدین شرح است:

1. su Si
   setfacl -m u:Sj:r O

2. su Si
   setfacl -m u:Sj:w O

3. su Si
   setfacl -x u:Sj:x O

4. su Si
   touch O
   chmod u=rwx O

الف) در دستور Create.file(Subj1, Obj4) ، چون یک Obj جدید ساخته می شود، یک ستون به ماتریس اضافه می‌شود و دسترسی های OWRX به فیلد مربوط به Subj1 در این ستون جدید مربوط به Obj4 اضافه می شود. پس ماتریس تغییر می کند.
در دستور Grant.write.file(Subj1, Obj3, Subj2) ، چون Subj1 مالک Obj3 نیست، پس این عملیات صورت نمی گیرد. پس ماتریس تغییر نمی کند.
در دستور Grant.read.file(Subj1, Obj4, Subj2) ، در فیلد مربوط بین Subj2 و Obj4 (که ستون نظیرش در کامند اول ایجاد شده بود و دسترسی ها به Subj1 داده شده بود)، دسترسی R اضافه می‌شود. پس ماتریس تغییر می کند.
در دستور Revoc.execution.file(Subj1, Obj3, Subj3) ، چون Subj1 مالک Obj3 نیست، پس این عملیات صورت نمی گیرد. پس ماتریس تغییر نمی کند.

ب) (فیلد خالی یعنی دسترسی ای ندارد.)

	Obj1	**Obj2**	**Obj3**	**Obj4**
Subj1	OWRX	R	RWX	OWRX
Subj2	R	RW	R	R
Subj3	RW	ORW	OWRX	

ج)

- ACL:

Obj1: {(Subj1, OWRX), (Subj2, R), (Subj3, RW)}

Obj2: {(Subj1, R), (Subj2, RW), (Subj3, ORW)}

Obj3: {(Subj1, WRX), (Subj2, R), (Subj3, OWRX)}

Obj4: {(Subj1, OWRX), (Subj2, R)}

- C-List:

Subj1: {(Obj1, OWRX), (Obj2, R), (Obj3, WRX), (Obj4, OWRX)}

Subj2: {(Obj1, R), (Obj2, RW), (Obj3, R), (Obj4, R)}

Subj3: {(Obj1, WR), (Obj2, ORW), (Obj3, OWRX)}

Day:..... Month:..... Year:.....

BLP

(4)

* خواندن O توسط S: $L(S)\ dom\ L(O)$
* نوشتن روی O توسط S: $L(O)\ dom\ L(S)$

الف) * خواندن: Top secret > Confidential

$\{A, B\} \not\subseteq \{B, C\}$

⊗ ← مجوز خواندن ندارد.

* نوشتن: Confidential ✗ top secret

$\{B, C\} \not\subseteq \{A, B\}$

⊗ ← مجوز نوشتن ندارد

ب) * خواندن: Secret ✗ top secret

⊗ ← مجوز خواندن ندارد.

* نوشتن: Top secret > secret

$\{A\} \not\subseteq \{B, C\}$

⊗ ← مجوز نوشتن ندارد

ج) * خواندن: Confidential ✗ secret

⊗ ← مجوز خواندن ندارد

* نوشتن: secret > Confidential

$\{A, B\} \not\subseteq \{A\}$

⊗ ← مجوز نوشتن ندارد

د) با توجه به اینکه مدل BLP را مخالفیم طرحه ایم، جریان اطلاعات از پایین به بالا است.

(a) (4)

Biba

* خواندن o توسط s: $i(o) \; dom \; i(s)$
* نوشتن روی o توسط s: $i(s) \; dom \; i(o)$

الف) * خواندن: very trusted > trusted

$\{B,C\} \not\subseteq \{A,B\}$

⟸ مجوز خواندن ندارد ✗

* نوشتن: trusted ✗ very trusted

⟸ مجوز نوشتن ندارد ✗

ب) * خواندن: Trusted > slightly trusted

$\{A\} \not\subseteq \{B,C\}$

⟸ مجوز خواندن ندارد ✗

* نوشتن: slightly trusted ✗ trusted

⟸ مجوز نوشتن ندارد ✗

ج) * خواندن: slightly trusted ✗ very trusted

⟸ مجوز خواندن ندارد ✗

* نوشتن: very trusted > Slightly trusted

$\{A\} \subseteq \{A,B\}$

⟸ مجوز نوشتن دارد ✓

د) با توجه به اینکه مخالفین مدل Biba را داشته باشیم و بیان اطلاعات از بالا به پایین است

Day..... Month..... Year.....

(5)

الف) + براساس پیاده سازی اش، ضعیف انعطاف دارد. وجی کوان کارهای مختلفی با آن انجام داد. مثلاً "نوع دسترسی (خواندن، نوشتن، ...) اینجا مشخص نیست. صریحاً "اگر یکی کلید داشته باشد، همه access ها را می تواند داشته باشد. ← اگر صریحاً "پیاده سازی رمز نگارانه را در نظر بگیریم، انعطاف حل و بررسی کنترل دسترسی اش کم است.

+ اگر چنین حالت رمز نگاری را در نظر بگیریم، بحث revocation این ضعیف ساده قابل پیاده سازی است. کافی است تمام کلیدهای رمز شده را از بین ببریم.

+ وی درباره delegation، کار ضعیف ساده نیست. یک نفر به سادگی می تواند کلید شش را در اختیار دیگران قرار دهد و دسترسی را به دیگران تفویض بدهد ← اگر از رمز نگاری کلید عمومی استفاده شود، این مسئله تا حدی قابل کنترل است. ولی باز هم شخص می تواند کلید خصوصی اش را در اختیار دیگران قرار دهد.

(5)

ب) * مقدار secret : 1523

* با توجه به اینکه معادله درجه 2 است ← threshold = 2+1 = 3

* $t=3$ نقطه داشته: (-1, 5313) (1, 7781) (0, 1523)

Day..... Month..... Year.....

$$l_0 = \frac{x-x_1}{x_0-x_1} \times \frac{x-x_2}{x_0-x_2} = \frac{x-1}{0-1} \times \frac{x+1}{0+1} = (1-x)(x+1) = x^2-2x+1$$

$$l_1 = \frac{x-x_0}{x_1-x_0} \times \frac{x-x_2}{x_1-x_2} = \frac{x-0}{1-0} \times \frac{x+1}{1+1} = \frac{1}{2}x(x+1) = \frac{1}{2}x^2+\frac{1}{2}x$$

$$l_2 = \frac{x-x_0}{x_2-x_0} \times \frac{x-x_1}{x_2-x_1} = \frac{x-0}{-1-0} \times \frac{x-1}{-1-1} = \frac{1}{2}x(x-1) = \frac{1}{2}x^2-\frac{1}{2}x$$

$$\Longrightarrow f(x) = y_0 l_0 + y_1 l_1 + y_2 l_2 =$$

$$= 1523(x^2-2x+1) + 7781\left(\frac{1}{2}x^2+\frac{1}{2}x\right) + 5313\left(\frac{1}{2}x^2-\frac{1}{2}x\right) =$$

$$= 1523(x^2-2x+1) + 13094\left(\frac{1}{2}x^2\right) + 2468\left(\frac{1}{2}x\right) =$$

$$= 1523x^2 - 3046x + 1523 + 6547x^2 + 1234x$$

$$a_0 = 1523$$

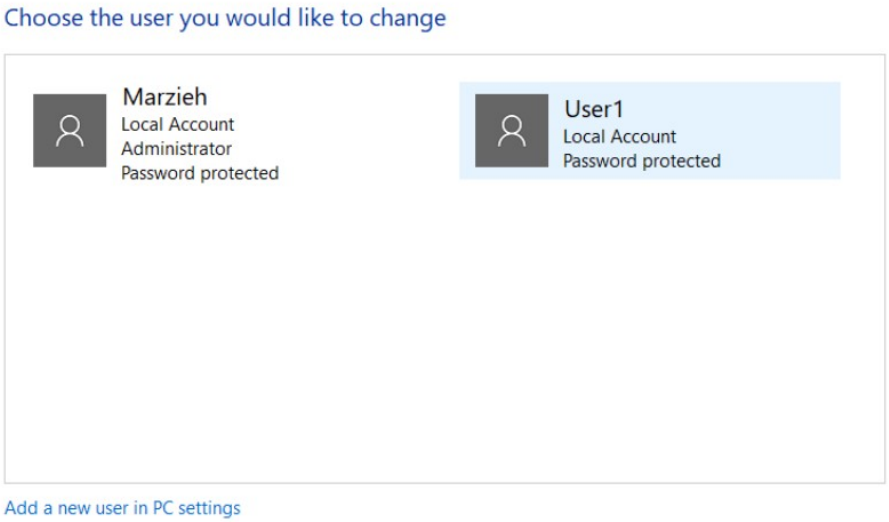
Secret = 1523 ← s_1

حال از این Secret به عنوان کلید رمزنگاری استفاده می‌شود و متن رمز شده باز می‌شود.

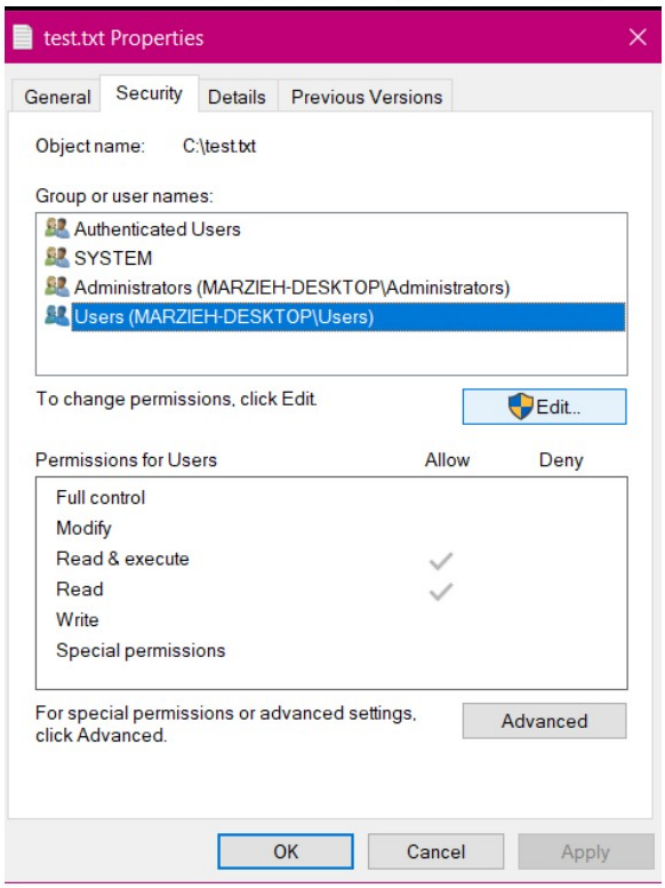
.7

الف)

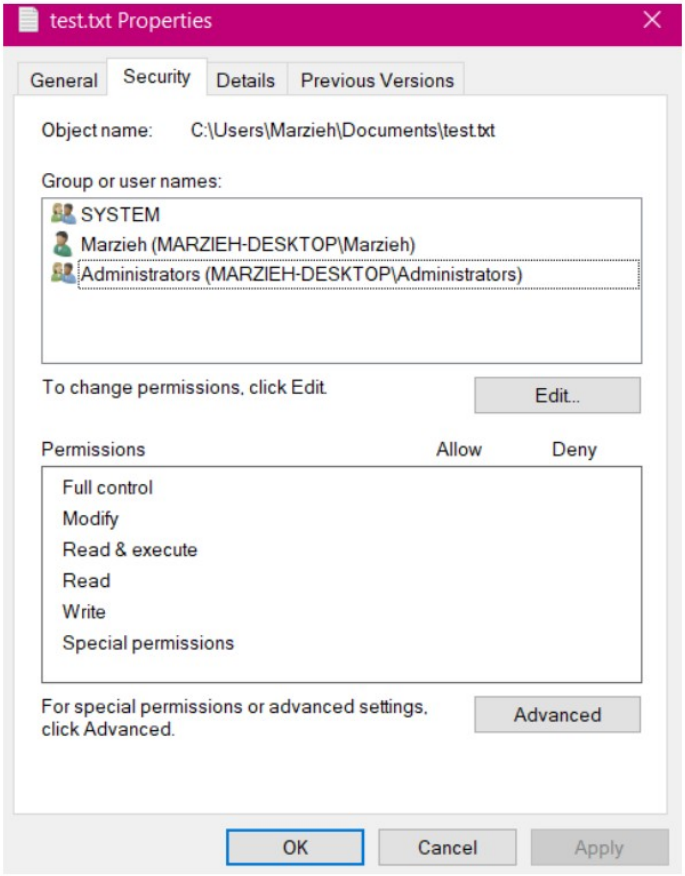
1. کاربر جدید:



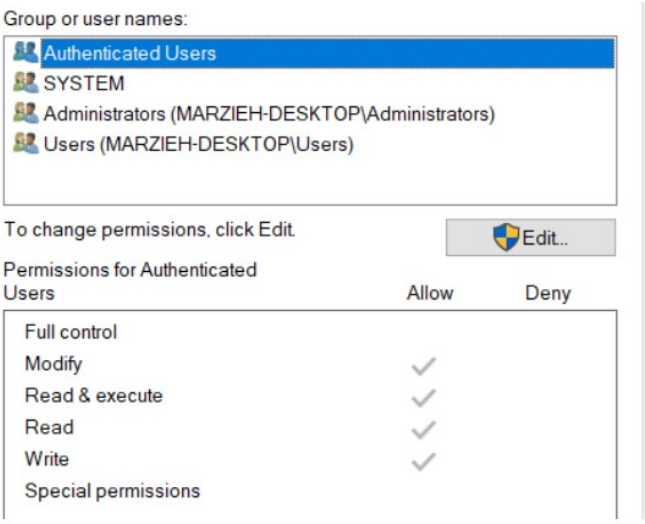
در حالتی که فایل در یک مسیر مشترک قرار دارد، همه ی user ها به یک اندازه به آن دسترسی دارند و فقط دسترسی خواندن و اجرا را دارند. پس User1 توانایی خواندن آن را دارد:



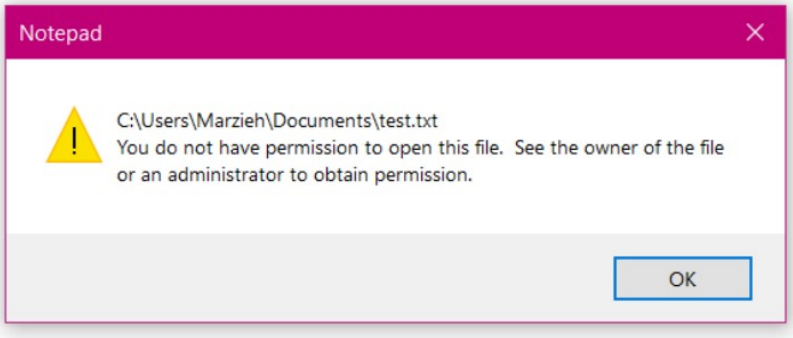
در حالتی که فایل در مسیر مربوط به کاربر جاری قرار دارد، User1 هیچ دسترسی ای ندارد و اصلا نام آن در لیست دسترسی ها آورده نشده است؛ که این نشاندهنده ی این است که اصلا دسترسی ای ندارد:



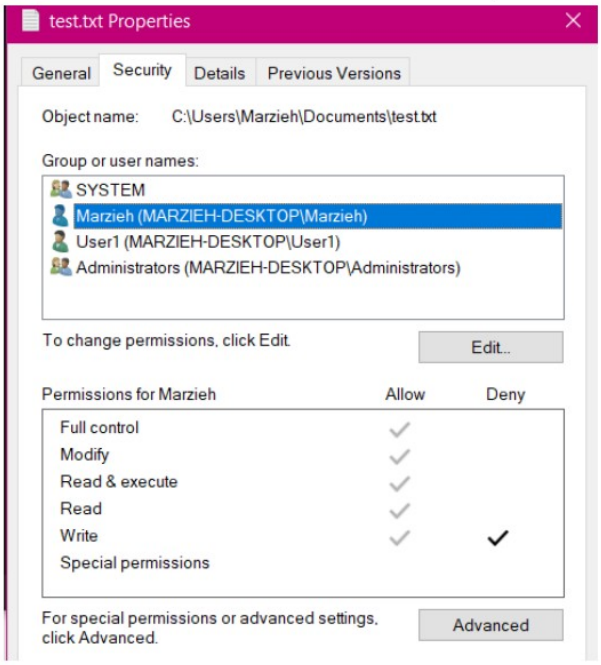
پیش فرض دسترسی های هر فایل به این صورت است که اگر دایرکتوری های مخصوص به یک کاربر قرار گرفته باشد، همه ی دسترسی ها به آن کاربر و به سیستم و root سیستم داده میشود؛ ولی به بقیه ی کاربران هیچ دسترسی ای تعلق نمی گیرد. ولی اگر فایل در مسیر مشترک کاربر ها در سیستم قرار گیرد، به سیستم و root آن دسترسی کامل تعلق می گیرد؛ به همه ی کاربران هم دسترسی خواندن و اجرا داده می شود؛ و به کاربران authenticate شده در سیستم با اسم و رمز هم دسترسی بیشتری تعلق می گیرد و فقط کامل نیست، که در شکل زیر آمده:



2.دسترسی های پیش فرض را برداشتم. و حالا کاربر جاری که فایل در مسیرش قرار دارد هم به فایل دسترسی ندارد:



به کاربر جاری دسترسی خواندن دادم:



به User1 هم دسترسی نوشتن دادم: