Информациска безбедност

Предавање:
Безбедност на оперативни системи

Проф. д-р Весна Димитрова

ОС и безбедност

- ОС - големи, комплексни програми
  - многу грешки во програмите
  - грешките можат да бидат закани за безбедноста
- Безбедносна заштита што ја нуди ОС
- Идеално:
  - кога ОС би можел да се справи со новонастанати ситуации кои влијаат на безбедноста на системот

ОС и безбедност

- Модерните ОС се дизајнирани за повеќе корисници и повеќе операции во исто време, па како резултат на тоа мора да можат да се справат со:
  - Memory, I/O devices (disk, printer, etc.), Programs, threads, Network issues, Data, etc.
- ОС мора да ги заштити процесите од други процеси и корисниците од другите корисници
  - Без разлика дали е случајно или злонамерно

Безбедносни функции на ОС

- Заштита на меморија
  - Ја заштитува меморијата од корисници/процеси
- Заштита на датотеки
  - Ги заштитува ресурсите на корисникот и системот
- Автентикација
  - Ги утврдува и спроведува резултатите за автентикација
- Авторизација
  - Ја одредува и спроведува контрола на пристап

Безбедносни функции на ОС

- Модерните ОС мора да можат да се справат со:
  - разделување,
  - мемориска заштита и
  - контрола на пристап.

Разделување

- Фундаментално прашање по безбедноста во модерните ОС е разделување.
- ОС мора да ги држи корисниците разделени еден од друг.
- Постојат неколку начини како може да се направи ова разделување:
- **Физичко разделување** каде корисниците се одвоени со посебни уреди.
  - Овој начин спроведува јака форма на раздвојување, но е чесно непрактичен.

Разделување

- **Привремено разделување**, каде што процесите се извршуваат еден по еден.
  - Ова ја поедноставува работата на ОС.
    - Едноставноста е пријател на безбедноста.
- **Логичко разделување** може да биде имплементирано преку sandboxing, каде што секој процес има свој sandbox.
  - Процесот може да прави што сака со неговиот sandbox, но е строго забрането да го прави тоа надвор од неговиот sandbox.
- **Криптографско разделување** ги прави информациите некорисни за надворешен член.

Заштита на меморијата

- Овој дел вклучува заштита на меморија што ја користи самиот ОС и меморијата што ја користат корисничките процеси.
- За заштита на меморијата може да се користи *fence* адреса.
  - Fence е одредена адреса која што корисниците и нивните процеси неможат да ја поминат и само оперативниот систем може да ја користи.
  - Значи оваа адреса наликува како еден вид на ограда која на едната страна го има само ОС, а на другата страна корисниците и нивните процеси.

Заштита на меморијата

- Fence адресата може да биде статична т.е фиксна fence адреса. Недостаток или може да биде и предност зависно како се гледа на тоа е тоа што фиксна fence адреса носи стриктен лимит на големината на ОС.
- Како алтернатива на ова може да се користи динамичка fence адреса која се имплеметира со употреба на fence регистар кој ќе ја специфицира актуелната fence адреса.

Заштита на меморијата

- Како дополнување на fence можат да се користат и базни и гранични регистри.
- Овие дополнителни регистри ги содржат најмалата и најголемата вредност на адресата на просторот кој може да го користи актуелниот корисник.
- Како ОС одлучува кој тип на заштита да се примени на специфична мемориска локација?

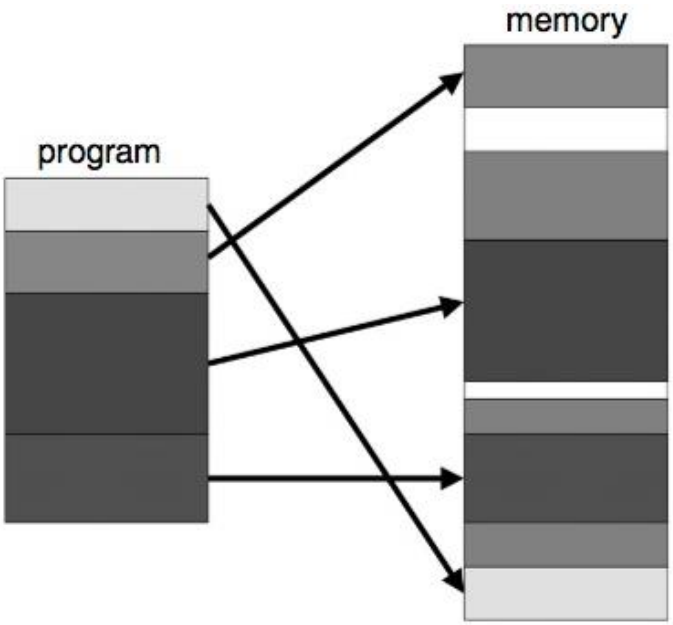
Заштита на меморијата

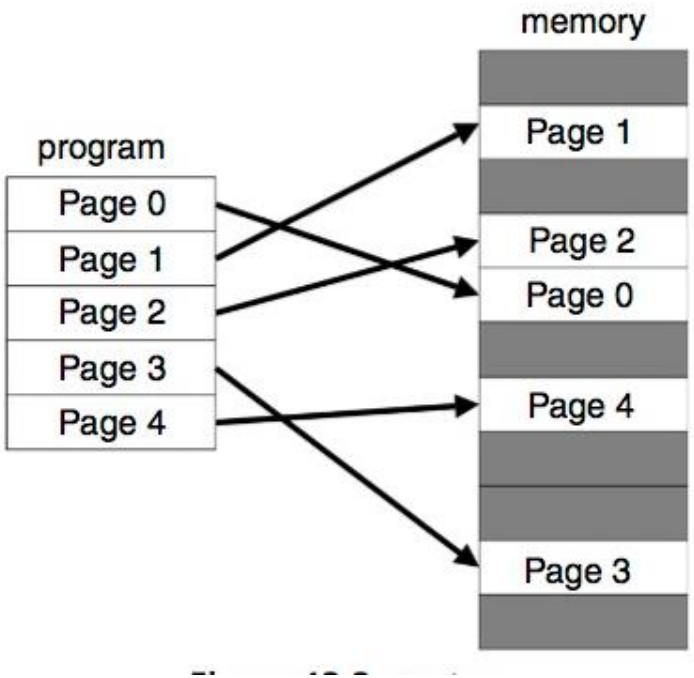
- Постојат два начини, а тоа е:
  - или да се примени иста безбедност за сите корисници
  - или да се примени tagging и во тој случај треба да се специфицира соодветна заштита за секоја адреса индивидуално.
- Иако вториот случај е далеку подобар, тој предизвикува големо оптеретување. Ова оптеретување може да се намали со тоа што место да се тагираат адресите може да се тагираат цели делови.

Заштита на меморијата

- Уште еден недостаток на тагирањето е проблем со компатибилноста, бидејќи овие шеми не се употребуваат толку често.
- Најкористени методи на заштита на меморијата се сегментација и страничење. Иако не се толку флексибилни како тагирањето, тие се далеку поефикасни.

Сегментација и страничење





Сегментација и страничење

- Сегментацијата како што се гледа на сликата ја дели меморијата во логички делови како индивидуални процедури или податоци во една низа. После тоа различни правила на пристап можат да се постават за различни сегменти.
- Друг бенефит од сегментирање е тоа дека секој сегмент може да биде ставен во било која мемориска локација - нормално ако локацијата е доволно голема за го прими сегментот.

Сегментација и страничење

- ОС мора да ги чува локациите на сите сегменти што се постигнува со употреба на (segment, offset) парот.
- Други предности на сегментирањето се тоа што може да се преместуваат на различни локации во меморијата и исто така може да се вадат и ставаат од и во меморија.

Сегментација и страничење

- Со сегментирање сите адресни референци мора да одат преку ОС, па во ваков случај ОС контролира се.
- Сегментирањето има еден сериозен недостаток, а тоа е дека сегментите се од различни големини.
- Како резултат на тоа кога ОС освен референцата (segment, offset) мора истотака да ја знае големината на сегментот за да биде сигурен дека побаруваната адреса е во рамки на сегментот.

Сегментација и страничење

- Некои сегменти - како тие што вклучуваат динамичка мемориска алокација - можат да пораснат за време на извршување.
- Според тоа ОС мора да ги чува сите големини на сегментите кои што можат да ја менуваат својата големина.
- Заради оваа променливост на големините на сегментите мемориската фрагментација е потенцијален проблем.

Сегментација и страничење

- Страничење е исто како сегментација со тоа што тука сите сегменти се со фиксна големина.
- Кај страничење пристапот до одредена страна вклучува пар од типот (page, offset).
- Предностите на страничење врз сегментирање вклучуваат:
  - Нема фрагментирање
  - подобрена ефикасност и
  - фактот дека веќе нема сегменти со различна големина
- Недостаток е тоа што нема логичка целина на страните и го прави потешко за доделување на соодветни правила за контрола на пристап.

Контрола на пристап

- Оперативниот систем ја спроведува контролата на пристап.
- Ова е една причина зашто оперативните системи се толку атрактивна мета на напаѓачите.
  - успешен напад на ОС ќе ја уништи заштитата на највисоко ниво.

Доверливи (Trusted) ОС

- Системот е доверлив ако можеме да се потпреме на безбедноста имплементирана во него.
- ОС е доверлив ако нуди:
  - Заштита на меморија
  - Заштита на датотеки
  - Автентикација
  - Авторизација

Доверливи (Trusted) ОС

- Ако доверлив систем не успее да ја овозможи очекуваната безбедност, тогаш безбедноста на целиот систем паѓа.
- Постои разлика помеѓу довеливост и безбедност.
- Доверливоста е бинарна операција т.е. системот е доверлив или не.

Доверливи (Trusted) ОС

- Од друга страна безбедноста е мерење на ефективноста на безбедносните механизми.
- Безбедноста се мери според претходно специфицирани полиси.
- Но безбедноста зависи од доверливоста, па идеално би било да се верува само во безбедни системи.

Доверливи (Trusted) ОС

- Оперативните системи посредуваат во интеракција помеѓу корисниците и ресурсите.
- Доверлив ОС може да користи разделување, заштита на меморијата и контрола на пристап.
- Тоа значи дека доверлив ОС мора да одлучи кои објекти ќе ги штити и како тоа ќе го направи, и исто така да одлучи на кои корисници што им е дозволено да прават.

MAC, DAC ...

- Било кој оперативен систем мора да овозможи некој степен на разделување, мемориска заштита и контрола на пристап.

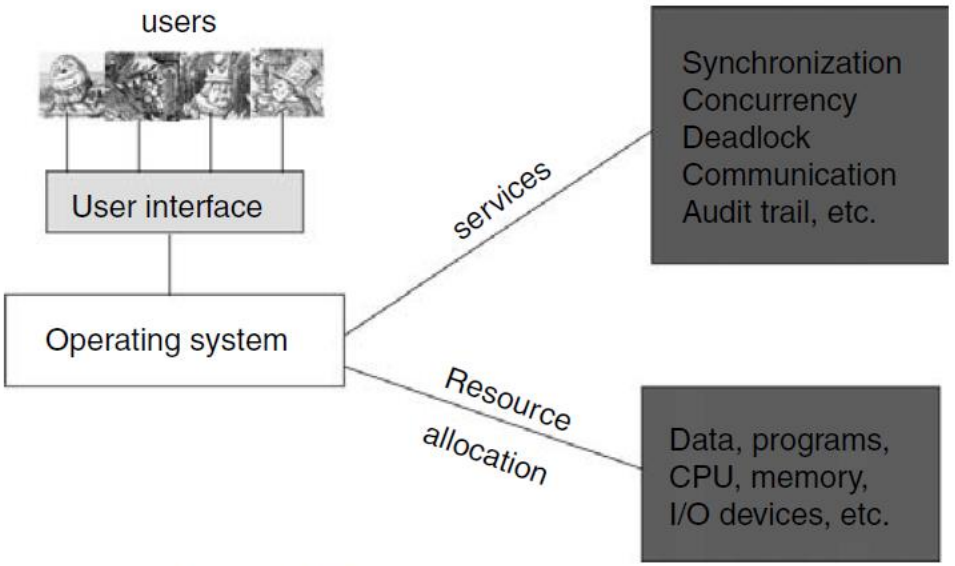


Figure 13.3. Operating system overview.

MAC, DAC ...

- Од друга страна, доверлив ОС мора да спроведе дополнителна безбедност вклучувајќи задолжителна контрола на пристап, дискретна контрола на пристап, заштита од повторно користење на објекти, комплетно посредување, доверлива патека и логови.

MAC, DAC ...

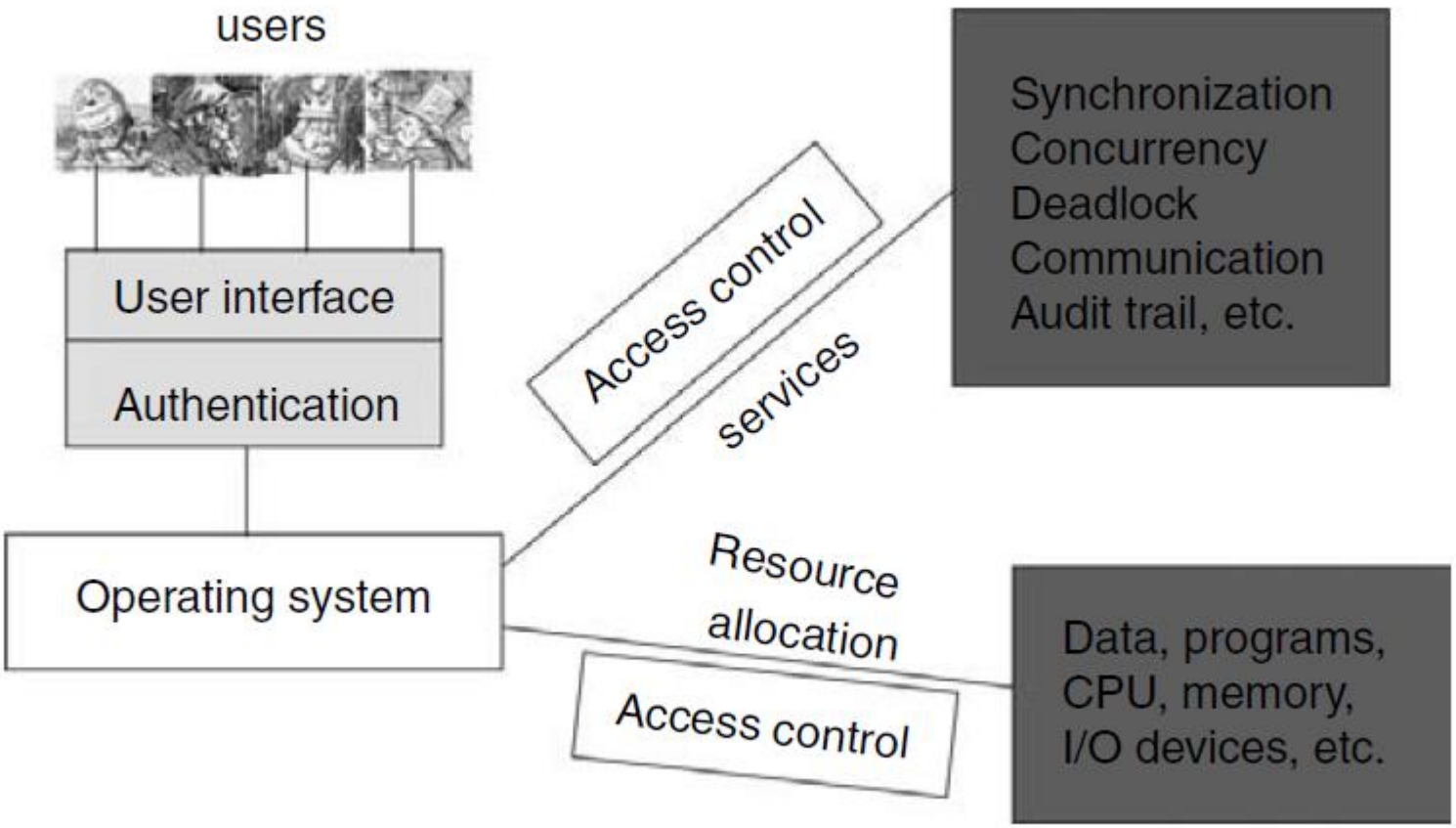


Figure 13.4. Trusted operating system overview.

MAC, DAC ...

- Задолжителна контрола на пристап (MAC) е пристап кој што не е контролиран од сопственикот на ресурсот.
- На пример Алис не одлучува кој има дозвола за пристап до TOP SECRET податоци, па таа неможе целосно да го контролира пристапот до документ на ова ниво.

MAC, DAC ...

- Дискретна контрола на пристап (DAC) е тип на контрола на пристап каде што пристапот се утврдува од страна на сопственикот на ресурсот.
- На пример заштитата на датотеки во UNIX, сопственикот на датотеката има контрола врз читај, запиши и изврши привилегии.

MAC, DAC ...

- Доверлив ОС мора истота така да спречи протекување на информации од еден корисник до друг.
- Секој ОС ќе користи некоја форма на заштита на меморијата и контрола на пристап за да спречи истекување на информации.

Доверлива патека

- Што се случува со лозинката која ја внесуваме при логирање во системот?
- Како може да бидеме сигурни дека софтверот не прави нешто лошо како на пример да ја запишува лозинката во текст фајл и подоцна да ја прати до некој малициозен корисник?
- Точно тоа прашање го решава "доверлива патека”.
- Идеално, доверлив ОС треба да овозможува јака гаранција за доверлива патека.

Доверлива патека

- Оперативниот систем е исто така одговорен за логирање настани поврзани со безбедноста.
- Овие инфромации се неопходни за детектирање на напад и за анализи.
- Логирањето не е толку едноставно како што изгледа.
- Ако се логираат премногу работи тогаш може да се затрупа цел систем, а и корисникот кој што ќе ги анализира тие логови.

Trusted Computing Base

- Јадрото (кернелот) е најниското ниво во ОС и е одговорен за синхронизација на комуникацијата помеѓу процесите, пренесување на пораки, ракување со прекини итн.
- Security kernel е дел од јадрото (кернелот) што кој што се занимава со безбедноста.
- Бидејќи секој пристап до некој ресурс мора да помине преку кернелот, тоа е идеално место за контрола на пристап.

Trusted Computing Base

- Добра пракса е да се стават сите безбедносни функции на една локација. Кога се сите функции на едно место тогаш нивото тестирање и модифицирање е многу полесно.
- Намерата на напаѓачите е да успеат да поминат под безбедносните функции на високо ниво и така да ја заобиколат безбедноста.

Trusted Computing Base

- Доколку овие безбедносни функции се стават во најниското ниво на ОС, тогаш ќе биде многу потешко за напаѓачот да ги заобиколи овие функции.
- Reference monitor е дел од безбедносниот кернел кој што се бави со контрола на пристап.

Trusted Computing Base

- Reference monitor посредува во сите пристапи помеѓу корисниците и ресурсите.

Figure 13.5. Reference monitor.

Trusted Computing Base

- Најбитниот дел од безбедносниот кернел треба да биде отпорен на промени и идеално би било да може да се анализира, бидејќи грешка на ова ниво ќе биде катастрофална за безбедноста на целиот систем.

Trusted Computing Base

- Trusted Computing Base или TCB е оној дел во ОС на кој што ние се потпираме за засилување на безбедноста на системот.
- Насекаде во оперативниот систем ќе се извршуваат операции важни за безбедноста.
- Идеално би било системот да се дизајнира така што ќе се почне од кернелот и после ќе се гради ОС врз основа на тоа.

Trusted Computing Base

- Но во реалноста е обично обратно, бидејќи безбедноста не е секогаш примарна цел во дизајнирањето на системот.
- Иако се ретки, постојат ОС кои што се дизајнирани со безбедност како примарна цел.
  - Таков пример е SCOMP безбеден оперативен систем развиен од Honeywell.
  - SCOMP има помалку од 10 000 линии код во неговиот безбедносен кернел и се стреми кон едноставност и можност за анализирање.

Trusted Computing Base

- Најдобро би било кога TCB би можело да ги собере заедно сите безбедносни функции во еден слој кој може да се идентификува