imoreno3 / **Homework-1-Moreno**

Branch: master ▾ **Homework-1-Moreno** / **homework1.md** Find file | Copy path

imoreno3 Update homework1.md c576885 on 14 Sep 2013

1 contributor

98 lines (86 sloc) 5.03 KB

"Homework 1"

#Chapter 1 **1. Classify each of the following as a violation of confidentiality, of integrity, of availability, or of some combination thereof.**

- **a. John copies Mary's homework.**
  - Cönfideǹţiaĺíţỳ
- **b. Paul crashes Linda's system.**
  - Aṽaiĺabiĺíţỳ
- **c. Carol changes the amount of Angelo's check from $100 to $1,000.**
  - Iǹţeǵřiţỳ
- **d. Gina forges Roger's signature on a deed.**
  - Iǹţeǵřiţỳ
- **e. Rhonda registers the domain name "AddisonWesley.com" and refuses to let the publishing house buy or use that domain name.**
  - Aṽaiĺabiĺíţỳ
  - Iǹţeǵřiţỳ

- **f. Jonah obtains Peter's credit card number and has the credit card company cancel the card and replace it with another card bearing a different account number.**
  - Cönfideṇţialĺţỳ
  - Iṇţeǵřiţỳ
  - Aṽaiĺabiĺiţỳ
- **g. Henry spoofs Julie's IP address to gain access to her computer.**
  - Iṇţeǵřiţỳ
  - Cönfideṇţialĺţỳ

3. The aphorism "security through obscurity" suggests that hiding information provides some level of security. Give an example of a situation in which hiding information does not add appreciably to the security of a system. Then give an example of a situation in which it does.

- Hidiṇǵ aṇ aĺǵöřiţ̈hḿ ţ̈haţ ṗröţecţš ỳöùř ṗaššẁörd ḿiǵ̈hţ ṇöţ ṇeceššařiĺỳ add aṗřeciabĺỳ ţ̈ö ţ̈he šecùřiţỳ öf ţ̈he šỳšţeḿ becùaše ţ̈he aĺǵöřiţ̈hḿ caṇ be föùṇd ẁiţ̈hiṇ ţ̈he šöùřce cöde öf ţ̈he Ĺibřařỳ*
- Höẁeṽeř#ḧidiṇǵ ţ̈he ṗaššẁörd fieĺd ẁiţ̈hiṇ a föřḿ ẁiĺĺ add aṗřeciabĺỳ ţ̈ö ţ̈he šỳšţeḿ*Ṫhiš ẁaỳ öṇĺỳ ţ̈he řiǵ̈hţfùĺ ùšeř ẁiĺĺ ḧaṽe accešš ţ̈ö ţ̈he accöùṇţ*

7. For each of the following statements, give an example of a situation in which the statement is true.

- **a. Prevention is more important than detection and recovery.**
- Přeṽeṇţiṇǵ ţ̈he acceššibiĺiţỳ öf a baṇḱ accöùṇţ iṇföřḿaţiöṇ
- **b. Detection is more important than prevention and recovery.**
- Deţecţiṇǵ ẁ̈heṇ aṇ eḿaiĺ iš šṗaḿ öř cöṇţaiṇš a ṽiřùš
- **c. Recovery is more important than prevention and detection.**
- Recöṽeřỳ föř daţa ceṇţeřš @öḿṗaṇieš, bacḱùṗš#

11. How do laws protecting privacy impact the ability of system administrators to monitor user activity?

- Laẁš ṗřöt̰ecṭiṅǵ ùšeř ṗřiṽacỳ caṅ šöḿeṭiḿeš be aṅ öbšṭacĺe föř adḿiṅišṭřaṭöřš ṭö dö a ǵööd ĵöb⁎Föř eẋaḿṗĺe⌗if aṅ ùšeř ẁeře ṭö be ĺeaḱiṅǵ iṅföřḿaṭiöṅ ṭö a cöḿṗeṭiṭöř ṽia eḿaiĺ aṅd ṭ̈he cöḿṗaṅỳ,š ĺaẁ döeš ṅöṭ aĺĺöẁ adḿiṅš ṭö c̈hecḱ föř eḿaiĺ cöṅfideṅṭiaĺ iṅföřḿaṭiöṅ⁎T̈heṅ ṭ̈he ùšeř ẁöùĺd ǵeṭ ẁaỳ ẁiṭ̈h ṅö ṗröbĺeḿ⁎Höẁeṽeř⌗if ṭ̈he adḿiṅ ẗaš accešš ṭö řead ṭ̈he iṅföřḿaṭiöṅ beiṅǵ ṭ̈řaṅšfeř⌗ṭ̈hiš cöùĺd be ṗřeṽeṅṭed⁎

#Chapter 2

1. Consider a computer system with three users: Alice, Bob, and Cyndy. Alice owns the file alicerc, and Bob and Cyndy can read it. Cyndy can read and write the file bobrc, which Bob owns, but Alice can only read it. Only Cyndy can read and write the file cyndyrc, which she owns. Assume that the owner of each of these files can execute it.

- **a. Create the corresponding access control matrix.**

	alicerc	bobrc	cyndyrc
Alice	--xo	r---	----
Bob	r---	--xo	----
Cyndy	r---	rw--	rwxo

- **b. Cyndy gives Alice permission to read cyndyrc, and Alice removes Bob's ability to read alicerc. Show the new access control matrix.**

	alicerc	bobrc	cyndyrc
Alice	--xo	r---	r---
Bob	----	--xo	----

	alicerc	bobrc	cyndyrc
Cyndy	r---	rw--	rwxo

2. Consider the set of rights {read, write, execute, append, list, modify, own}.

- **a. Using the syntax in Section 2.3, write a command delete_all_rights (p, q, s). This command causes p to delete all rights the subject q has over an object s.**

- command delete_all_rights(p, q, s)
  delete read from a[q, s];
  delete write from a[q, s];
  delete execute from a[q, s];
  delete append from a[q, s];
  delete list from a[q, s];
  delete modify from a[q, s];
  delete own from a[q, s];
  end

- **b. Modify your command so that the deletion can occur only if p has modify rights over s.**

- command delete_all_rights(p,q,s)
  if modify in a[p,s] then
  delete read in a[q,s]
  delete write in a[q,s]
  delete execute in a[q,s]
  delete append in a[q,s]
  delete list in a[q,s]
  delete modify in a[q,s]
  delete own in a[q,s]
  end

- **c. Modify your command so that the deletion can occur only if p has modify rights over s and q does not have own rights over s.**

- command delete_all_rights(p,q,s)
  create subject tmp
  enter read in a[tmp,s]
  if own in a[q,s] then
  delete read from a[tmp,s]
  if modify in a[p,s] and read in a[tmp,s] then
  delete read in a[q,s]
  delete write in a[q,s]
  delete execute in a[q,s]
  delete append in a[q,s]
  delete list in a[q,s]
  delete modify in a[q,s]
  delete own in a[q,s]
  destroy subject tmp
  end