Computer Security Tutorials

Computer Security COMPGA01

Nicolas T. Courtois, November 2011

Answers to be filled in directly with a pen, solutions are given in class. The question numbers are of the form X.Y where X corresponds to the numbering of pdf slides. Many questions are the same as in previous exam or homework.

Question 2.1.

Question	Answer
Explain what is a *security policy*.	A security policy is a short, succinct statement. It is written at a high-level and describes what is and what is not allowed.
A *security mechanism* (in this context).	A mechanism by which a security policy can be implemented or enforced.
What is a reference monitor?	A reference monitor is a (most of the time a software) module that controls all software access to data objects, files, devices, I/O ports etc.
State the three main properties it should satisfy.	It should be i) tamper-proof, ii) always-invoked = non-bypassable (complete mediation), and iii) simple (economical + small enough to be build in a rigorous way, and fully tested and analysed).
Explain DAC and MAC. Which one is more vulnerable to malware?	Discretionary Access Control : people to grant rights at their Discretion. In practice users grant their privileges to programs they run, and these privileges will be substantial and exploited by Trojans. Mandatory Access Control rules are mandatory, and simply forbid certain operations or certain information flows, limiting damage from malware.
Explain briefly the Chinese wall model in terms of classes, information flows, transitive closure in directed graphs.	It defines "conflict of interest classes". If two firms, say Pepsi and Coca-Cola are in the same "conflict of interest class" all potential **information flows** that (through transitivity) can **potentially** (a strict policy) occur and lead to data flowing from one of these firms to another, will be prevented.

Question 2.2.

Question	Answer
What is an order relation in mathematics? What are the R A T axioms?	Order relation: A set A , relation R Reflexive, Antisymmetric, Transitive. $\forall \in A$ aRa, etc...
What is a POSET?	A poset (Partially Ordered SET) is a set with an order relation. GLB = Greatest Lower Bound, a.k.a. Inf. Defined as the greatest element dominated by the arguments.
What is the dual notion for LUB? Alternative names?	GLB = Greatest Lower Bound, a.k.a. Inf or Meet or Wedge $\wedge$.
Exact mathematical definition?	Defined as the greatest element dominated by the arguments. $GLB(a,b) = Max_{\leq}\{x \in A \mid x \leq a \text{ AND } x \leq B\}$
In $< \mathbb{N}, \mid >$, what is the transitive closure of $\mid$?	This relation is transitive, so its transitive closure is itself, the relation $\mid$.
Given a set of classifications H with a total ordering $\leq_H$, and a lattice $C = P(Cat), \subseteq$ where Cat is any set of "categories", write a definition of the Bell-LaPadula product lattice.	The Bell LaPadula product lattice is defined as a poset $H \times C, \leq$ where $\leq$ is defined by: $(l_1, c_1) \leq (l_2, c_2) \iff l_1 \leq_H l_2 \wedge c_1 \subseteq c_2$

Question 2.3. Consider the set of confidentiality levels

$$L = \{PUB < OFF < SEC\},$$

and the set of categories

$$Cat = \{Production(P), HumanRessources(H), Finance(F)\}.$$

We consider four objects o_1, o_2, o_3, o_4 and two subjects s, t with the following clearance levels:

$$\begin{aligned} \lambda(o_1) &= (PUB, \{P\}) \\ \lambda(o_2) &= (PUB, \{\}) \\ \lambda(o_3) &= (OFF, \{H, P\}) \\ \lambda(o_4) &= (SEC, \{P\}) \\ \lambda(s) &= (OFF, \{H, F, P\}) \\ \lambda(t) &= (SEC, \{P\}) \end{aligned}$$

Answer the following questions in the Bell LaPadula model:

Question	Answer
Count security classes in this classification lattice.	$3 \cdot 2^3 = 24$
The Bottom element $\bot =$	$(PUB, \{\})$
The Top element $\top =$.	$(SEC, \{P, H, F\})$
LUB$(\lambda(o_1), \lambda(o_4)) =$	$(SEC, \{P\})$
which users can read both o_1 and o_4?	t
GLB$(\lambda(o_1)), \lambda(o_3)) =$	$(PUB, \{P\})$
which users can write both o_1 and o_3	both if connected as $(PUB, \{P\})$
LUB$(\lambda(s)), \lambda(t)) =$	$(SEC, \{P, H, F\}) = \top$
which object can be written by both s, t	none
GLB$(\lambda(s)), \lambda(t)) =$	$(OFF, \{P\})$
which objects can be read by both s, t	o_1, o_2
Which objects s can read.	o_1, o_2, o_3
Which objects t can write?	o_4

Question 2.4. Consider the set of integrity levels

$$L = \{UserSpace(US) < System(SH)\}.$$

Consider the following set of categories

$$Cat = \{SensitiveWorkFiles(S), PersonalData(P)\}.$$

Let Bob be a subject and and $\{do1, fi2, pr3\}$ a set of objects with the following classifications.

$$\begin{aligned} \lambda(Bob) &= (SH, \{S\}) \\ \lambda(do1) &= (US, \{P\}) \\ \lambda(fi2) &= (SH, \{S\}) \\ \lambda(pr3) &= (US, \{S\}) \end{aligned}$$

Fill in the following table working all the way down for each of the 5 policies. Consider that the operations are executed in order, so that potential changes in security levels λ can affect further operations.
We recall that LWM = Low Water Mark policy. In the strict Biba and in the Ring policy current levels never change. In the Integrity Audit policy all operations are permitted and only changes in λ need to be recorded.
For each operation (working column by column) do explain whether the operation will be allowed (Y) or denied (N). Note any potential changes (if any) to the security classes: write a new value of $\lambda(x)$ each time it is changed. If it does not change, there is no need to write it.

policy ▷	Biba strict	LWM for Objects		LWM for Subjects		Integr. Audit	Ring
operation	Y/N	Y/N	λ change	Y/N	λ change	λ change	Y/N
read(do1)	N	N		**Y**	$\lambda(Bob) = (\mathbf{US}, \{\})$	$\lambda(Bob) = (\mathbf{US}, \{\})$	**Y**
read(pr3)	N	N		**Y**	$\lambda(Bob) = (US, \{\})$	$\lambda(Bob) = (US, \{\})$	**Y**
write(fi2)	Y	Y	$\lambda(fi2) = (SH, \{S\})$	**N**		$\lambda(fi2) = (\mathbf{US}, \{\})$	Y
write(do1)	N	Y	$\lambda(do1) = (\mathbf{US}, \{\})$	N		$\lambda(do1) = (\mathbf{US}, \{\})$	N
write(pr3)	Y	Y	$\lambda(pr3) = (US, \{S\})$	**N**		$\lambda(pr3) = (\mathbf{US}, \{\})$	Y
read(do1)	N	N		**Y**	$\lambda(Bob) = (US, \{\})$	$\lambda(Bob) = (US, \{\})$	**Y**
write(fi2)	Y	Y	$\lambda(fi2) = (SH, \{S\})$	**N**		$\lambda(fi2) = (US, \{\})$	Y
write(do1)	N	**Y**	$\lambda(do1) = (US, \{\})$	N		$\lambda(do1) = (US, \{\})$	N
write(pr3)	Y	Y	$\lambda(pr3) = (US, \{S\})$	**N**		$\lambda(pr3) = (US, \{\})$	Y
read(pr3)	N	N		**Y**	$\lambda(Bob) = (US, \{\})$	$\lambda(Bob) = (US, \{\})$	**Y**

Question 3.1. Here is a listing of a Unix directory.

Permissions	Owner	Group	Size	Last Update	File Name
`-rws-----x`	dave	gdev	1452306	Nov 03 21h11	gtool
`drwxrwxrwt`	dave	gdev	1452306	Nov 03 21h11	gdata
`-rwx--x--x`	alice	alice	214768	Nov 03 09h36	setup
`-rw-r-----`	alice	pcrack	12486	Dec 04 11h00	sourcg
`-rw-r--r--`	dave	pcrack	14257	Oct 02 18h44	config
`-rw--wxr--`	root	pcrack	176704	Nov 01 12h23	hosts

Suppose that user alice is a member of groups alice and pcrack. User dave is a member of groups dave, pcrack, and gdev. For each question specified in the following table, provide your responses.

Question	Answer
List the names of the files that alice can write.	setup; sourcg; hosts;
List the names of the files that dave can read.	sourcg; gtool; config;
Suppose that alice executes program gtool. List the names of the files that the corresponding process can execute.	setup; gtool; hosts;
Suppose that dave executes program setup. List the names of the files that the corresponding process can write.	gtool; config; hosts;
How do we distinguish directories? Explain x permission for directories.	First letter d for gdata. Means one can CD to that dir, and traverse a directory to access subdirectories.
The permissions for gtool start with -rws. Explain what does 's' stand for?	Setuid permission. Program will have the access rights of the owner of the file, even if another user is running the process. the uid can also be changed during the execution.
Explain what sticky bit is. when does it apply.	Directories only, letter t or T ix x present. Last group. Only the owner of the file, or owner of the directory, or root can remove or rename files contained in gdata.

Question 3.2.

Question	Answer
In Unix/Windows, can a process be more privileged than the user who calls it? Give one example.	For example a program that is able to change your password. It makes changes in data files that ordinary users are not able to read. And in many systems the administrators cannot access the password file either.
Can it be less privileged? One example?	If the owner is less privileged.
Explain what is *Real User Id* and *Effective User Id* in Unix systems.	The Real User Id (ruid) identifies the owner of the process, the Effective User Id determines current access rights and can change during the execution of the process, for example to drop certain privileges.
Explain very briefly how in Windows, a system knows if a user is allowed to access a specific file with (Discretionary) Access Control Lists (ACL's).	Each object/file has a security descriptor, which a list of Access Control Elements (ACE) objects. Each ACE says that for some user or group (or "local system", or another type of SID = Security Identifier) a certain set of rights is allowed.

Explain what a "closed" policy is. In Apache web servers, explain what happens when the *.htaccess* file contains the following 3 lines in order.
`Order Allow,Deny`
`Deny from all`
`Allow from cs.ucl.ac.uk`
	Deny by default, and deny overrides allow. Here it overrides it even when some allow is specified. The current configuration file with "`Deny from all`" and "`Allow from ucl.ac.uk`" will produce a somewhat very strange outcome: no one will be able to access the web site(!). Tricky question.

Question 5.1.

Question	Answer
*Which is bigger $H(X,Y)$ or $H(X)$? When equality is achieved?	$H(X,Y) \geq H(X)$. Equal if and only if Y depends on X
*Which is bigger $H(X,Y)$ or $H(X)+H(Y)$? When equality is achieved?	$H(X,Y) \leq H(X)+H(Y)$. Equality if and only if X and Y are independent.
For a discrete variable with n outcomes which is bigger $H(X)$ or $log_2(n)$? When equality is achieved?	$H(X) \leq log_2(n)$. Equality if and only if X is uniform.
*Which is bigger $H(X\|Y)$ or 0? Equality?	$H(X,Y) \geq 0$. Equal if and only if Y is a function of X
*Which is bigger $H(X\|Y)$ or $H(X,Y)-H(Y)$? When equality is achieved?	Always equal. $H(X\|Y) = H(X,Y) - H(Y)$.
*Which is bigger $H(X\|Y)$ or $H(X)$? When equality is achieved?	$H(X\|Y) \leq H(X)$. Equality if and only if X and Y are independent.
Define Entropy of a password with distribution $p_1, \ldots, p_n$.	$-\sum_{i=1}^{n} p_i \cdot log_2 p_i$
In which case the entropy measures the strength of a password?	For one single user / target.
Define Min-entropy of a password.	$-log_2$ P(most frequent password)
In which case the Min-entropy measures the strength of a password?	For a large number of users / targets, breaking at least one.

Question 5.2.

Question	Answer
What is "spoofing" in the context of password security?	Fake login page.
What are the three factors? Why writing the password down defeats a 2-factor system without necessarily making it less secure? Solutions?	Now the password becomes also something we have. Solution: write some part of the password.
Give two examples of self-defeating security recommendations regarding passwords.	Make passwords longer → users write them down. Change passwords → users worry more not to remember them and make them less secure.
Can passwords be possibly stored encrypted by a deterministic block cipher algorithm with a fixed key?	No, the key should be different for each user.
What is the encryption AND the storage is implemented in a secure hardware?	Yes, it will work. No attack.
How to use a hash function to store a password?	Store $H($ name, salt, machine ID, password), salt