ISAE Embedded systems master

Evaluation – Exercices and questions – WITH CORRECTION

7 february 2017

Computer security

Advice to students and supervisors : course documents (either furnished by the school or hand-written during oral courses by the student himself) are allowed during the examination, a standard calculator too (for calculation purposes only) and blank paper sheets for draft.

All other documents or media access are not allowed, unless direct explicit authorization from the session supervisor.

Please, write your answers on the document itself in the place reserved.

All explanations are given in good faith and in italics without any guarantee for applicability for any purpose. Remember the best student is not the one who has the better mark, but the one who can reach any mark he/she aims.

Part I (10 pts)

This first part consists of ten questions (1 pt per question) with multiple answers proposed among which you must select the appropriate one. Unless explicitly indicated, only one answer is the right one.

Attention, the following notation system will be used :

Right answer : 1 point added

False/no answer : 0 point

Q1 What are currently the most efficient incentives for top level management with respect to security?

- ☐ [✓ Legal *(or regulatory)* constraints.]
- ☐ Consumer demand.
- ☐ Economical constraints.
- ☐ Company reputation.

Explanation : Consumer demand and economical constraints are typical not very strong with respect to security at the moment. Company reputation may be more important, but depends on the area. Legal (or regulatory) constraints are those most referred to.

Q2 Which of these algorithms is suitable to establish *(may benefit from)* a public key directory ?

- ☐ AES
- ☐ RSA
- ☐ SHA-3
- ☐ DES

Explanation : RSA is the only public key algorithm mentionned.

Q3 What is the security development method needed to reach the highest evaluation levels in normalized evaluation criteria :

- ☐ A trusted execution path (SysRq).
- ☐ The abilitiy to select between a discretionay or mandatory security policy at login
- ☐ A formal proof of (a model) of the security kernel functions and its implementation
- ☐ Presidential signature on the certificate.

Q4 Error codes must always be checked after calling a library function because :

- ☐ most programming books recommend them ;
- ☐ all software developpers do that all the time ;
- ☐ it will prevent the program from crashing ;
- ☐ it is the only way of preventing abuse of API misuse interactions.

Q5 Which step of the application development phase is quasi systematically omitted even from security-oriented computer software development :

- ☐ Test funding.
- ☐ End-user security need.
- ☐ Data disposal.
- ☐ Developper holiday.
- ☐ Authentication delegation.

Q6 What is the methodological information provided by a CERT/CC (Computer Emergency Response Team) ?

- ☐ Examples of awareness raising documents for internal communication toward employees
- ☐ Career development guidelines for computer security officers
- ☐ Secure software development rules for various programming languages
- ☐ Secure contract rules for export control conformance

Q7 Given that the total number of atoms in the universe is usually estimated around 10^82, what is the incentive for selecting a 2^256 bit key length instead of a 2^128 bits key length for AES :

- ☐ Because, you never know, your adversary may have access to several universes to attack you.
- ☐ Because someone may already have broken the 2^128 bit key length version but not the extended one.
- ☐ Because you can and the additional energy cost is marginal.
- ☐ Because it will cost more and motivate the development of commercial encryption devices.

Q8 Which habilitation is allowed to access a document of security classication (CONFIDENTIAL, {NAVY, TECHNICAL, RADAR}) under the Bell-La Padula security policy (and the natural ordering of labels) :

- ☐ (TOP SECRET, { AIR FORCE, SALARIES })
- ☐ (SECRET, { NAVY, TECHNICAL, ENGINE})
- ☐ (PUBLIC, {NAVY})
- ☐ (CONFIDENTIAL, { NAVY, AIR FORCE, TECHNICAL, LOGISTICS, RADAR, SONAR})
- ☐ (CONFIDENTIAL, { NAVY, AIR FORCE, LOGISTICS, RADAR, SONAR})

Q9 Before a competitive exam, the computer on which all results are to be consolidated and sorted is locked in a glass walled room visible by everyone, the programs to run are audited and checked and the official publication place of the results is decided and announced to the candidates. This kind of procedure is related to :

- ☐ confidentiality
- ☐ integrity
- ☐ availability
- ☐ or *survivability*

Q10 In the above configuration, the easiest avenue for an attacker (the residual vulnerability) to disrupt and discredit the whole exam is :

- ☐ by intercepting and altering the commmunication channel between the computer and the publication medium to display funny results.
- ☐ by studying very hard to rank first in the competitive exam and then publicly despising how « easy it was ».
- ☐ by breaking into the computer room and stealing the computer.
- ☐ by intercepting and altering correction reports from professors.
- ☐ by intercepting exam questions and selectively leaking them to a significant fraction (e.g. 15%) of the candidates (but not all).

Explanation : Answer 1 and 2 are only perturbation of the publication phase. Answer 3 and 4 may necessitate re-doing the consolidation of reports and the final compilation, but in the fifth case, the equality of chances of all candidates has been irrevocably compromised (and will certainly not remain secret given the number of favored ones) which will probably necessitate redoing it entirely.

Q11 In a networked sensors system relying on the Biba multilevel mandatory integrity policy, is it possible that a low integrity level CPU uses a high integrity sensor output in order to perform a computation ?

- ☐ YES
- ☐ NO

Explanation : The computation result will be of a low integrity level however.

Part II (10 pts)

This part is composed of five open questions (2 pts each). Please write down your answer *on this document* in the appropriate space.

Question 1

Consider an automatic drone delivery system (for conventional goods). Propose 4 security objectives of this system.

2 of them corresponding to the needs of the distributor using the delivery system :

Prevent anyone from destroying the drones in order to grab the goods for free.

Prevent consumers from denying being delivered .

Prevent anyone from disrupting drone in order to damage some installation.

Prevent anyone from taking control of the drone to pillage inventory.

Prevent anyone from stealing drones.

And the 2 others to the needs of the end customer of the delivery system :

Prevent the delivered good from being damaged during delivery.

Prevent the drone from being told to select another delivery place.

Prevent the drone from blocking the flying car.

Prevent the good from being identified during delivery.

Question 2

Propose 4 programming rules for enhancing the security of a C software development project.

Perform regular checks of heap integrity.

Use calloc() instead of malloc() for memory allocation of storage areas for complex items.

Forbid strcpy() usage.

Printf() should always provide a format.

Allocate sensitive ressources at initialization and revoke most privileges before accepting user input.

Read https://www.securecoding.cert.org/confluence/display/c/SEI+CERT+C+Coding+Standard for further applicable rules[1].

Question 3

Give 4 examples of security vulnerabilities affecting informations systems (*at least* one in each of the hardware, software and organizational category).

(hardware/physical) Easy monitoring and dispersion of WiFi radio waves

(hardware) Networking switches fallback to broadcast when switching tables are exhausted

(software) Arithmetic rounding errors

(software) Lack of data input validation (e.g. for env. variables)

(software) Execution of input data

(organizational) Transmission of personal password

(organizational) Transmission of paper signed order by (unsigned) email

(organizational) Delegation of data storage resources

[1]Note the self extensibility of the rule...

Question 4

Give examples of the 4 different approaches to risk management :

Risk avoidance

Project cancellation

Authentication delegation to a network service

Removal of vulnerable programs from the system

Risk reduction

Software code audit checking

Addition of secure hash generation and checking function for critical data

Development of replacement programs.

Risk acceptance

Asteroid impact on the data center

Simultaneous accidental death of all customers

Risk transfer

Reliance on an independent X.509 certification authority for identity checking

Data recovery costs insurance

Question 5

Here are several ideas for entirely removing buffer overflow problems. Discuss their adequacy (do they work) and applicability (do they sound realistic).

Do not use function calls, but only coroutines (aka. jumps).

Buffur overflows primarily come from C calling conventions, so a coroutine programming style should allow to prevent such abuses.

However, such a programming style may also be prone to vulnerabilities. For example, if adresses calculations are used to find a coroutine target.

Furthermore, this programming style is not really familiar to most programmers.

However, for control software, it may be appropriate, especially if the final code is generated from some other (e.g. specification level) tool.

Change the CPU architecture to have a (second) separate stack for storing return adresses.

Apart from the fact that it necessitates to develop a new alternate CPU architecture, it sounds like a good idea for specifically eliminating buffer overflows.

Cost is probably going to be the limiting factor here, as well as the transition of all current computers to this new CPU ; not counting the fact that we could also take the opportunity to adresse a few other common software vulnerability causes at the same time.

Make the stack non-executable and more generally prohibits self-modifying code.

NX stacks are the common route taken to prevent the most basic classes of buffer overflow. Software (OS or compiler-level) solutions may be proposed for this, or more reduced hardware improvements that are starting to become common.

The general prohibition of self modifying code is more a programming methodology concern, but seems to be a realistic target also.